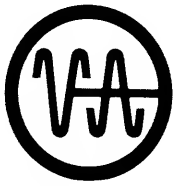


varian data machines /a varian subsidiary

DATA 620/i INTERFACE REFERENCE MANUAL

DATA 620/i
INTERFACE
REFERENCE
MANUAL

varian data machines /a varian subsidiary
2722 michelson drive / irvine / california / 92664 / (714) 833-2400

98 A 9902 014
October 1968

CONTENTS

ILLUSTRATIONS

TABLES

SECTION 1
INTRODUCTION

1.1 GENERAL DESCRIPTION

The DATA 620/i computer is a high-speed parallel, binary computer. Its extensive instruction repertoire, flexible input/output, and modular packaging make it ideally suited for application as a general-purpose machine or as an on-line system component.

Its features include:

Fast operation	1.8-microsecond memory cycle.
Large instruction repertoire	107 standard plus 18 optional, with approximately 200 micro instructions.
Word length	16- or 18-bit configurations.
Modular memory	4096 words. Expandable to 32,768 words maximum.
Multiple addressing modes	Six, including direct, indirect, relative, index, immediate and extended (optional).
Flexible I/O	10 device controllers on standard I/O bus; optional interlaced input/output.
Extensive software	Programming and diagnostic aids are provided for efficient system use.
Modular packaging	Processor and 4096-word memory module occupy only 10-1/2 inches of rack height; additional memory module requires only 10-1/2 inches additional.

The DATA 620/i is simple in design and is easy to program, operate and maintain. As a system component, it is easily integrated with other equipment through the use of standard or special peripheral interface elements. The computer and its associated power supplies and peripheral controllers all mount in standard 19-inch equipment cabinets, and require no special cabling or air conditioning facilities.

1.2 PURPOSE OF THIS MANUAL

This manual provides basic circuit and logic design, and timing information on the standard and optional input/output (I/O) facilities of the DATA 620/i, plus design examples for several I/O functions. Using this information, the system designer can integrate the computer with special interfaces tailored to specific system requirements.

This manual also contains information on cabling and grounding, thus serving as a basic document for system planning purposes. While a detailed knowledge of the internal computer is not essential for successful interface design, it is recommended that the system designer have a general familiarity with the computer organization and operation. The available publications for the DATA 620/i are summarized in table 1-1.

Table 1-1
DATA 620/i Technical Publications

Publication Number	Title
98 A 9902 002	System Reference Manual
98 A 9902 023	Programming Reference Manual
98 A 9902 031	FORTRAN Reference Manual
98 A 9902 041	Subroutine Descriptions
98 A 9902 050	Maintenance Manual
98 A 9902 100	Mainframe Options Manual
98 A 9902 300	Installation and Integration Manual

1.3 COMPUTER ORGANIZATION

The overall organization and basic information paths of the DATA 620/i are shown in figure 1-1. The basic system is composed of the following functional elements: control section, arithmetic/logic section, operational register section, input/output section and memory section. An optional input/output facility, direct memory access, is also available.

1.3.1 Control Section

The control section decodes the program instructions into timing and control signals for the entire machine. There are 107 standard instructions decoded; an additional 18 instructions may be supplied as options. Approximately 200 microcoded instructions may be derived from the standard instruction set.

1.3.2 Arithmetic/Logic Section

This section contains the gating elements required to perform all programmed arithmetic and logic operations. It is also used for internal control operations such as instruction and operand address modification.

1.3.3 Operational Register Section

Operational registers include the A, B, X and P registers. The A and B registers form a double-length register for arithmetic and logical operations. The B register may also be used for indexed addressing. The X register is a full word-length hardware index register. Indexed addressing using the B or X register requires no additional time for execution of the instruction. Registers A and B may also be used for direct input/output transfers. The instruction counter, P, holds the memory address of the instruction being executed by the control section. The S bus provides routing of these registers to the arithmetic unit.

1.3.4 Input/Output Section

This section provides transmission of control and data signals to and from peripheral device options attached to the I/O cable (a peripheral device option consists of a peripheral controller that controls one or more peripheral devices). The I/O-bus interface with the computer control and register sections is shown in figure 1-1.

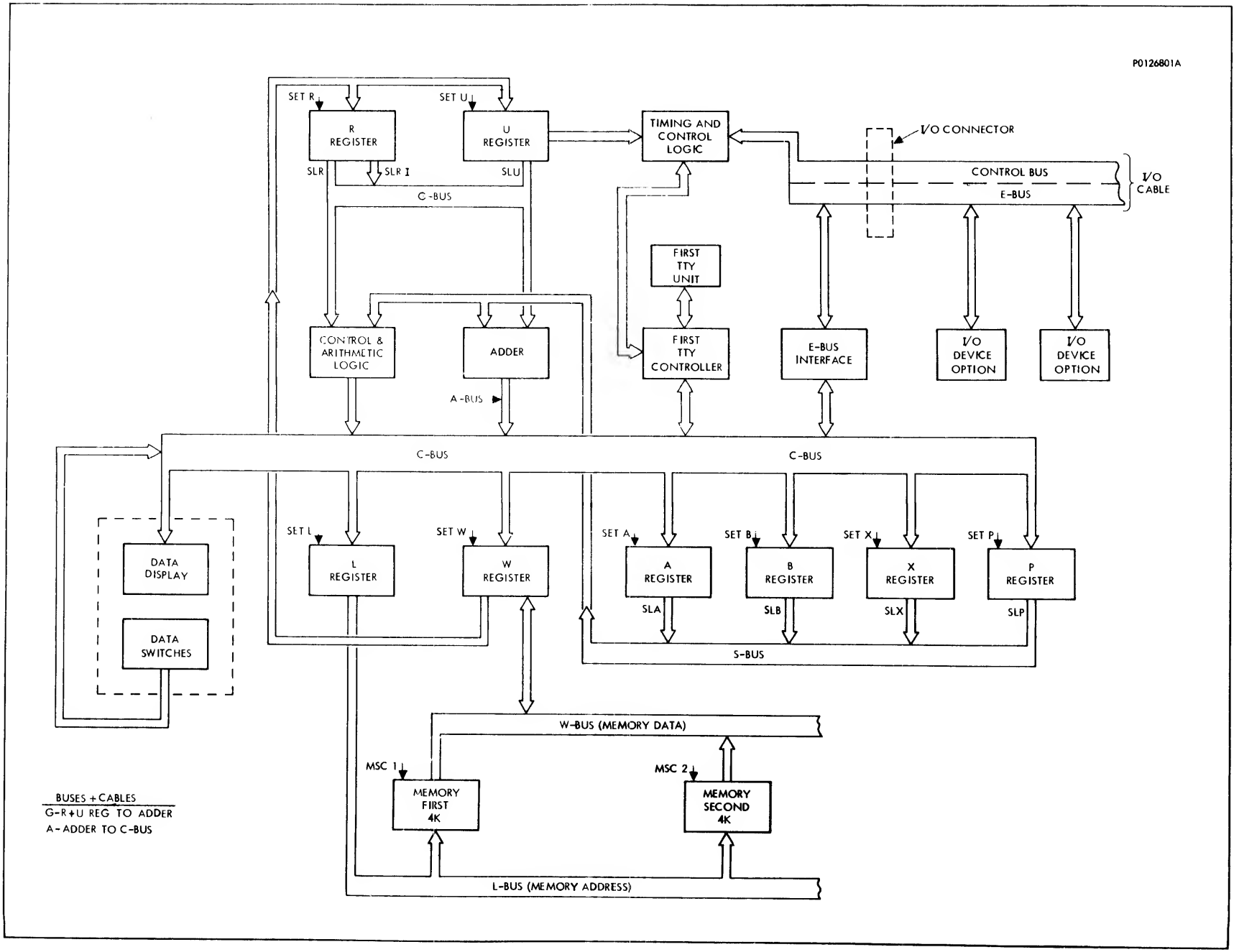


Figure 1-1. 620/i Organization

A total of 64 peripheral device addresses is available (drive capability for ten controllers is standard). External program sense and interrupt functions are transferred to and from the control section through the I/O section. Data transfers may be single-word (program controlled) or block (using the optional buffer interlace controller and direct-memory-access feature).

1.3.5 Memory Section

Memory modules are slaved to the computer, which contains the address and data registers for all modules. Minimum memory size is 4096 words. Total memory may be expanded to 32,768 words in increments of 4096 words.

SECTION 2
DATA 620/i INPUT/OUTPUT SYSTEM

2.1 GENERAL DESCRIPTION

The standard computer, without options, communicates with peripheral equipment on a word-parallel (either 16- or 18-bit) input/output (I/O) bus as shown in figure 2-1. Each information transfer occurs under the control of a stored program. Information exchanges with peripheral devices are synchronized by peripheral controllers. A controller may control one or more similar peripheral devices. Each controller and the device(s) it controls comprise a peripheral device option.

The basic machine provides four types of I/O operations:

External Control	An external control code is transmitted under program control from the computer to a peripheral controller.
Program Sense	The status of a selected peripheral controller sense line is interrogated by the computer under program control.
Single-Word Input Transfer	A single word of data is transferred under program control from a peripheral controller to the A register, B register or any location in memory.
Single-Word Output Transfer	A single word of data is transferred under program control to a peripheral controller from the A register, B register or any location in memory.

2.1.1 Direct-Memory-Access-and-Interrupt Option

A computer equipped with the direct-memory-access (DMA/I) option is capable of I/O data transfers to or from memory without program intervention. DMA/I allows external devices on the I/O bus to request data transfers on a priority basis while temporarily interrupting processing of the stored program. The program is caused to remain idle for 3.15 microseconds during which the transfer takes place. This "cycle stealing" action does not disturb the operational registers (A, B, X, P), thus allowing the program to proceed normally at the conclusion of the data transfer.

The interrupt feature of the DMA/I option permits an external device to request execution of an instruction that is independent of the main program. When two or more requests are received simultaneously, the hardwired external device priority scheme determines which device has priority. The computer is directed to a memory location specified by the interrupting device and is caused to execute the instruction found at that location. Any DATA 620/i instruction other than an I/O command may be executed in this manner. Normally, however, the instruction is a jump-and-mark instruction and results in the processing of an I/O subroutine.

2.2 BASIC I/O BUS SIGNAL INTERFACE

A computer equipped with the standard I/O interface can communicate directly with all peripheral device options under program control. The computer may initiate operation of a peripheral device by transmitting an external control code and a proper device address to the selected controller via the I/O bus. The computer may determine when a device is ready to send or receive information by interrogating its associated sense line. A device may be requested to place a word of data on the I/O bus during a computer input transfer or to accept a word of data placed on the bus by the computer during an output transfer.

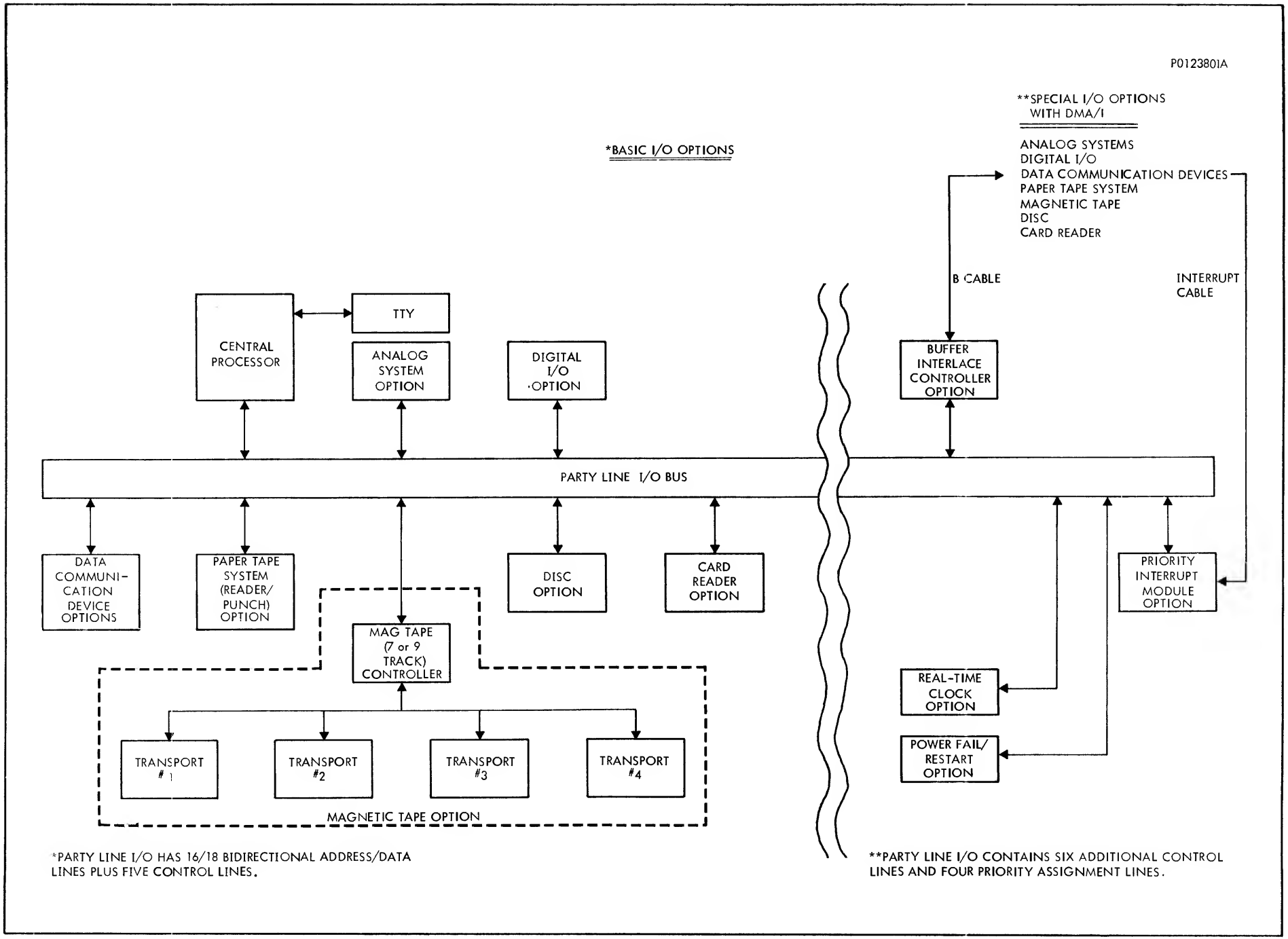


Figure 2-1. DATA 620/i Input/Output System

The standard I/O bus without options consists of the E bus and five I/O control lines: FRYX-I, DRYX-I, SERX-I, IUAX-I and SYRT-I.

2.2.1 E Bus (EB00-I through EB17-I)

The E bus is a 16- or 18-bit, parallel, bidirectional I/O channel. It is used to transmit control codes, device addresses and data from the computer to the peripheral device options. In turn, the bus is used by these options to transmit data to the computer. A total of ten drivers and ten receivers may be connected to each line. An E-bus signal is logically true when it is at 0 vdc and is logically false when it is at +3 vdc. A typical E-bus configuration is shown in figure 2-2.

2.2.2 FRYX-I

Signal FRYX-I is a pulse generated by the computer to indicate that the computer has placed a device address and a control code on the E bus. Each peripheral controller examines the device address and, upon the true-to-false transition of FRYX-I, the addressed device responds to the control code. FRYX-I is logically true at 0 vdc and is logically false at +3 vdc. A total of ten receivers may be connected to the line. A typical FRYX-I line configuration is shown in figure 2-3.

2.2.3 DRYX-I

Signal DRYX-I is a pulse generated by the computer. During an output data transfer, DRYX-I indicates that the computer has placed data on the E bus and that the peripheral device addressed previously should strobe the data into its input buffer. During an input data transfer, DRYX-I indicates that the computer has accepted the data placed on the E bus by the peripheral device and that, following the true-to-false transition of DRYX-I, the device should remove the data. DRYX-I is logically true at 0 vdc and is logically false at +3 vdc. A total of ten receivers may be connected to the line. A typical DRYX-I line configuration is shown in figure 2-3.

2.2.4 SERX-I

During the execution of a program sense (SEN) command, the computer places a function code and a device address on the E bus. The addressed controller is instructed to indicate the status of the specific device condition that is identified by the function code. If the specified condition is true, the controller responds by setting the SERX-I line true. If the condition is false, SERX-I is left false. SERX-I is logically true at 0 vdc and is logically false at +3 vdc. A total of ten drivers may be connected to the line. A typical SERX-I line configuration is shown in figure 2-4.

2.2.5 IUAX-I

Signal IUAX-I is generated by the computer to acknowledge that either a DMA/I access or interrupt operation is in progress. Each peripheral device controller uses IUAX-I to inhibit normal device address decoding. In a basic I/O interface, without the DMA/I option, IUAX-I is held false (+3 vdc). A total of ten receivers may be connected to the line. A typical IUAX-I line configuration is shown in figure 2-3.

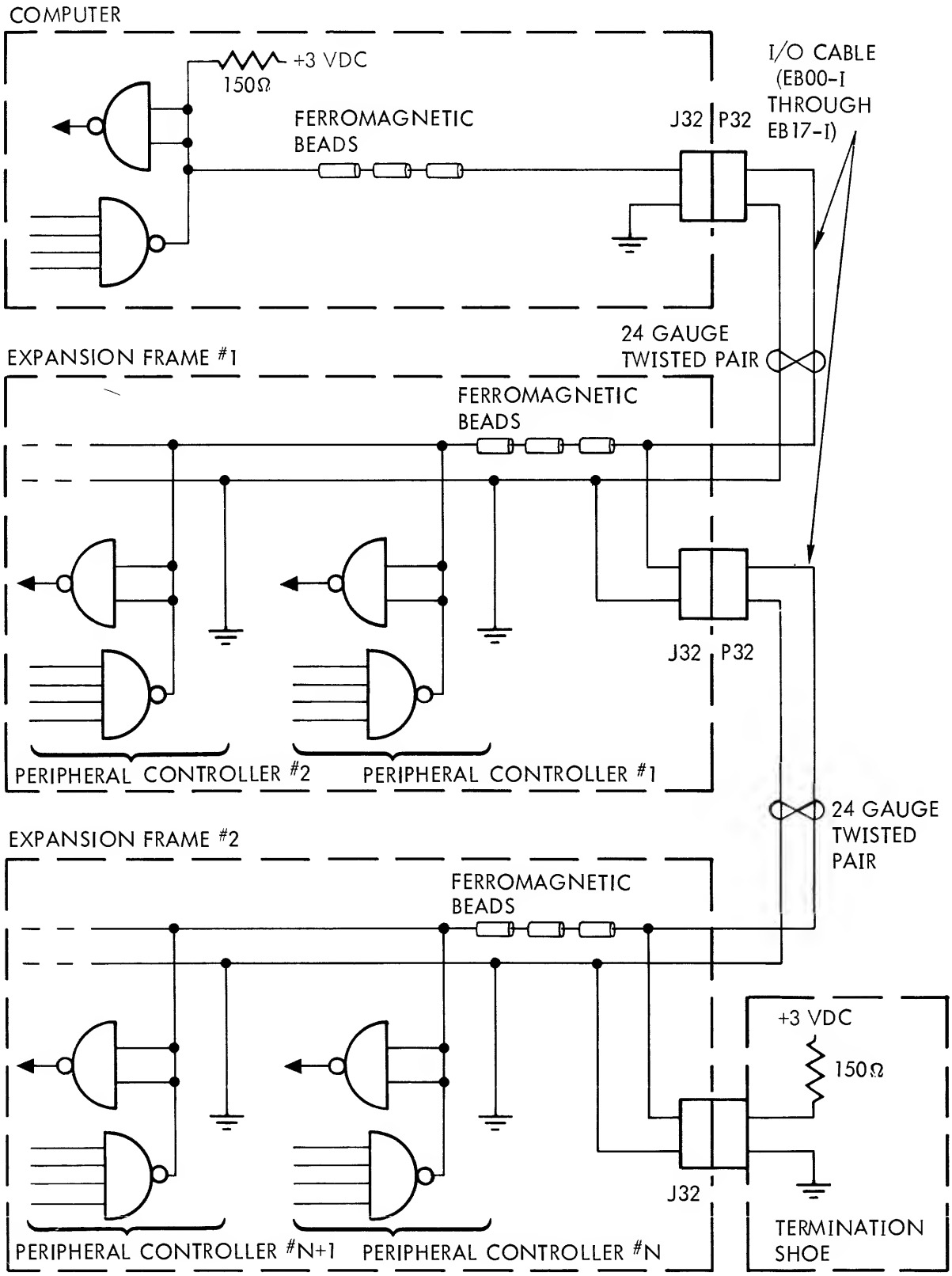


Figure 2-2. Typical E-Bus Line

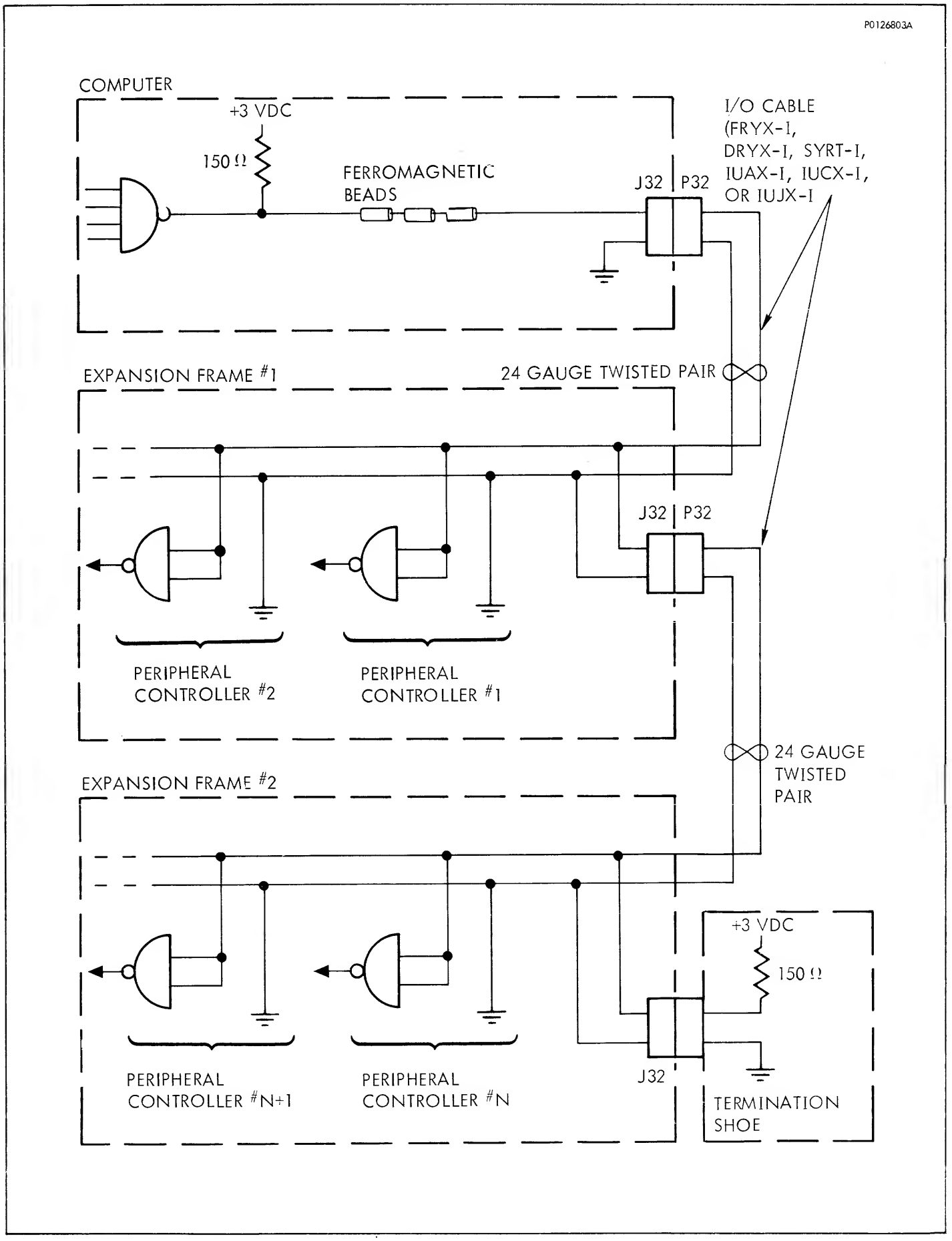


Figure 2-3. Typical Control Signal From The Computer

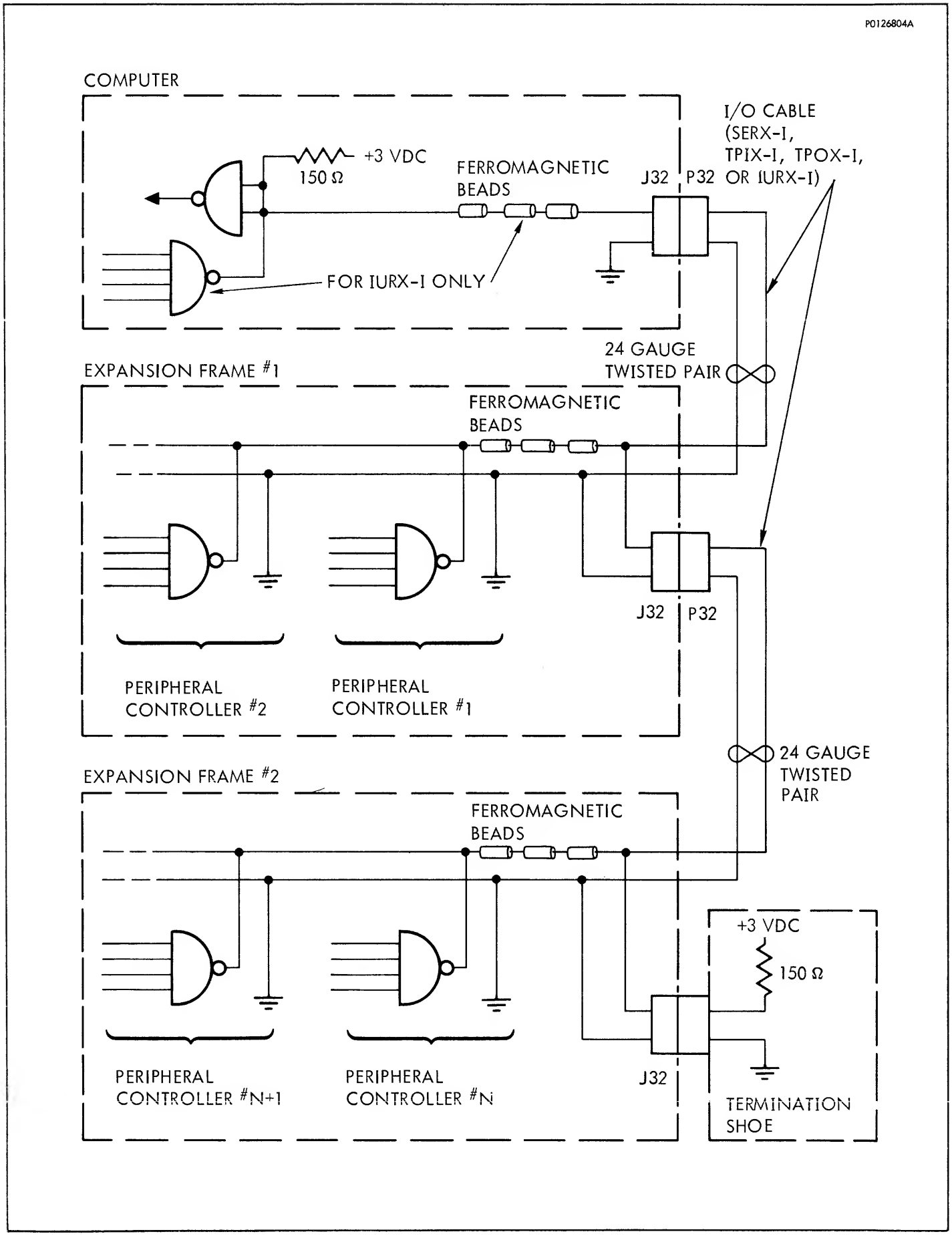


Figure 2-4. Typical Control Signal To The Computer

2.2.6 SYRT-I

Signal SYRT-I is used to initialize each peripheral device controller connected to the I/O bus. SYRT-I becomes true when the SYSTEM RESET switch on the control console is pressed. SYRT-I is logically true at 0 vdc and is logically false at +3 vdc.

A total of ten receivers may be connected to the line. A typical SYRT-I line configuration is shown in figure 2-3.

2.3 I/O COMMAND OPERATIONS

The basic computer provides four types of I/O commands for program control of peripheral devices connected to the I/O bus. These are:

a. External control.

b. Program sense.

c. Single-word input transfer.

d. Single-word output transfer.

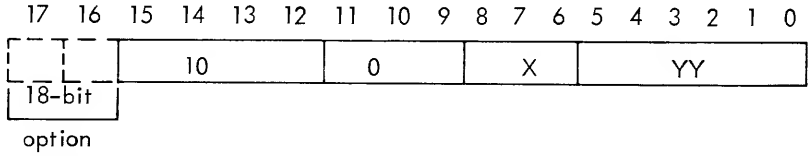
Table 2-1 summarizes E-bus and control signals used during these commands.

The following paragraphs describe the operations associated with each command type and the manner in which the I/O-bus signals are used. The E-bus signals used for each command execution are shown in table 2-2.

Device codes have been assigned to the standard DATA 620/i peripheral devices, as shown in table 2-3. All device codes are in the range 00_8 to 77_8. Each peripheral device belongs to a class, according to its function. Each class is assigned a block of codes, and specific code assignments are given to devices in that class.

2.3.1 External Control

The external control (EXC) command is used to initiate a specific mode of operation in a peripheral device. An example is the use of an EXC command to cause a magnetic tape transport to advance the tape one record. The EXC instruction word format is shown below, where YY contains the device address and X contains the function code.



The EXC command causes the function code and device address portions of the instruction word to be placed on the E bus. The EXC I/O timing is shown in figure 2-5. Signal lines EB00-I through EB05-I indicate the device address; EB06-I through EB08-I indicate the function code. EB11-I is held true, indicating that an external control function is being performed. The device controller decodes the binary function code and, following the true-to-false transition of FRYX-I, initiates the specified mode of operation in the addressed device. During the executionof EXC, no data are exchanged between the computer and the device controller, and no response signal is expected from the controller.

Table 2-1
E-BUS AND I/O CONTROL SIGNALS

	Operation						
	External Control	Sense	Data Transfer (Single Word I/O)		Trap Sequence (Buffer Interlace Control)		Interrupt Sequence
					TPOX-I or TPIX-I		
Control Lines	FRYX-I* (Phase 1)	FRYX-I* SERX-I* (Phase 1)	FRYX-I* (Phase 1)	DRYX-I (Phase 2)	IUAX-I, FRYX-I (Phase 1)	IUAX-I, DRYX-I (Phase 2)	IURX-I IUAX-I (Phase 1)
E-Bus Meaning							
EB00-I to EB05-I	Device Address	Device Address	Device Address	Data Being Transferred In or Out	Memory Address In	Data In or Out	Use lines 01-15 for interrupt location by pairs.
EB06-I to EB08-I	Function Code	Function Code	Not Used				
EB09-I EB10-I	Not Used	Not Used	Not Used				
EB11-I	External Control Command						
EB12-I		Sense Command					
EB13-I			Data Transfer In				
EB14-I			Data Transfer Out				
EB15-I	Extended External Control Command						

*IUAX Interlock - used in the address decoding term.

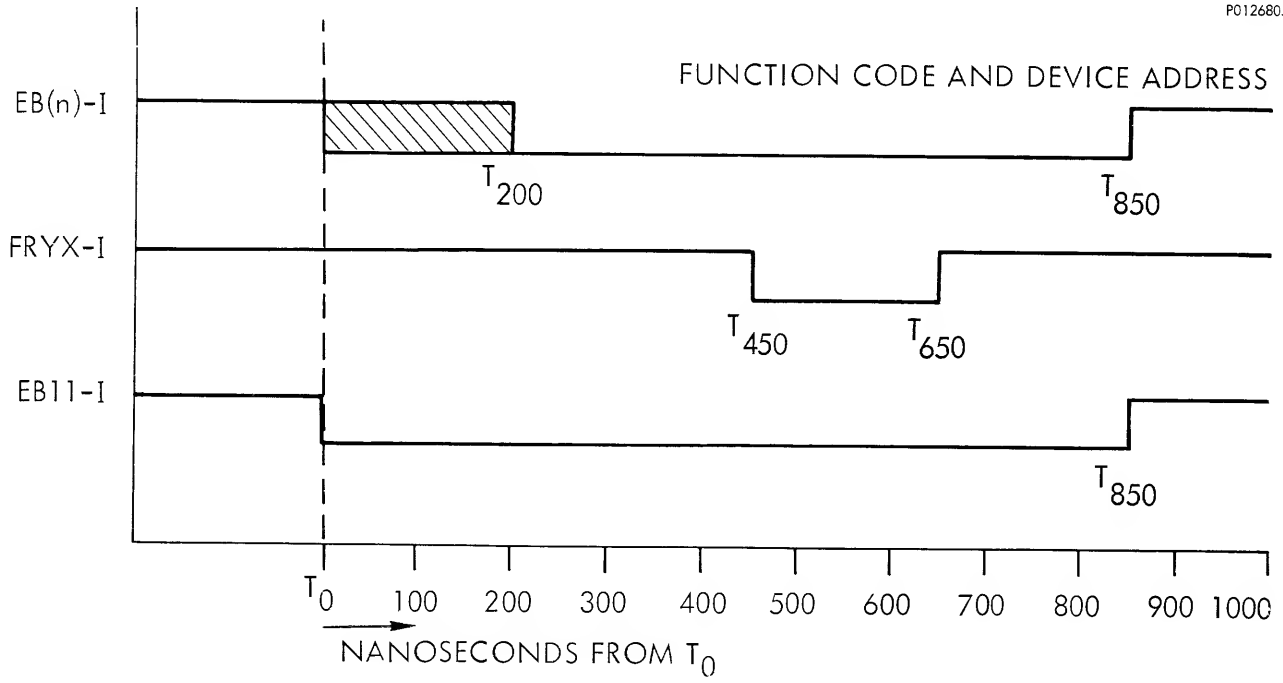
Table 2-2
E-Bus Signals

OPERATION	EB16-I and EB17-I	EB15-I	EB14-I	EB13-I	EB12-I	EB11-I	EB10-I, EB09-I	EB08-I	EB07-I	EB06-I	EB05-I to EB00-I
Send External Control Code to External Device		0	0	0	0	1	Unused	Function Code			Device Address ($00-63_{10}$)
Sense State of External Device		0	0	0	1	0	Unused	Function Code			Device Address ($00-63_{10}$)
Input to Memory								0	0	0	
Input to A Register		0	0	1	0	0	Unused	Note 1	0	1	Device Address ($00-63_{10}$)
Input to B Register								Note 1	1	0	
Output from Memory								0	0	0	
Output from A Register		0	1	0	0	0	Unused	Unused	0	1	Device Address ($00-63_{10}$)
Output from B Register								Unused	1	0	
Send Extended External Control Code to External Device		1	0	0	0	0	Unused	Function Code			Device Address ($00-63_{10}$)

Note 1: If EB08-I is true, selected register in computer is cleared before input.
If EB08-I is false, selected register in computer is not cleared. The result after input is the logical -OR of the original register and the input signals.
Bits EB06-I to EB08-I are ignored by the I/O controller during data transfers.

TABLE 2-3
DATA 620/i Standard Device Codes

Octal Class Codes	Octal Addresses Assigned	Peripheral or Option
00-07	00-07	ASR-33/35 Teletype Controller (serial/parallel)
10-13	10,11	Magnetic Tape Controllers
14-17	14-17	Disc Controllers
20-27	20,21	First Buffer Interlace Controller
	22,23	Second Buffer Interlace Controller
	24,25	Third Buffer Interlace Controller
	26,27	Fourth Buffer Interlace Controller
30-37	30	Card Reader
	31	Card Punch
	32	Digital Plotter
	35,36	Line Printers
	37	Paper Tape System (punch and/or reader)
40-47	40	Priority Interrupt Module
	45	Memory Protect
	47	Real-Time Clock
50-53	50-53	Optical Devices
54-57	54-57	Analog Systems
60-67	60-67	Digital Input/Output Controllers
70-77	70-77	Special Applications



T_0 is the start of the execute phase of the external control instruction.

Logic levels: true = 0 VDC
false = +3 VDC

= time when signal is settling.

Figure 2-5. External Control Timing

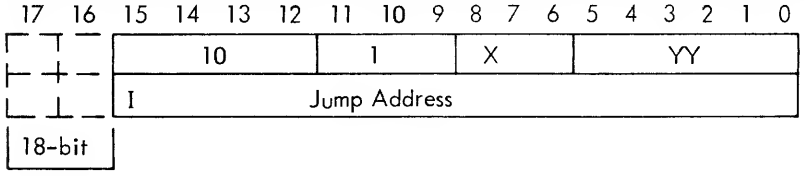
When additional function codes are required, the extended EXC command (instruction code 10 4) may be used. This command is identical to the normal EXC command (instruction code 10 0) in timing and function, but is identified by EB15-I instead of EB11-I as shown in tables 2-1 and 2-2.

Figure 2-6 shows typical implementation of the logic required in a peripheral controller to receive, decode and perform an EXC command.

2.3.2 Program Sense

The program sense (SEN) command is used to test the status of a specific device condition and, if a true condition is detected, a program jump is made. If a false condition is detected, the next instruction in sequence is executed. An example of SEN command usage is a test to determine if a magnetic tape transport is rewinding.

The SEN instruction format is shown below, where YY contains the device address and X contains the function code, which defines the specific condition to be tested.



The SEN command causes the function code and device address portions of the instruction word to be placed on the E bus. The SEN I/O timing is shown in figure 2-7. Signal lines EB00-I through EB05-I indicate the device address. EB12-I is held true to indicate that a SEN command is in progress.

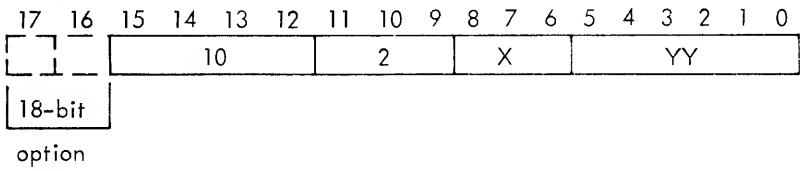
Figure 2-8 shows typical implementation of the logic required in a peripheral controller to receive, decode and respond to a SEN command. Note that the device address (usually ANDed with IUAX-I) can be used directly to enable the sense line response (SERX-I). EB12-I need not be used to enable SERX-I, since the computer samples the SERX-I line only when a SEN command is executed.

2.3.3 Single-Word Input Transfer

There are five instructions which provide a single-word input transfer. These are:

a. Input to A register (INA)*.
b. Input to B register (INB)*.
c. Input to memory (IME).
d. Clear and input to A register (CIA).
e. Clear and input to B register (CIB).

The instruction word format for INA, INB, CIA and CIB is shown below, where YY contains the device address and X defines each individual command.



*Inclusive OR of the data lines and the specified register contents.

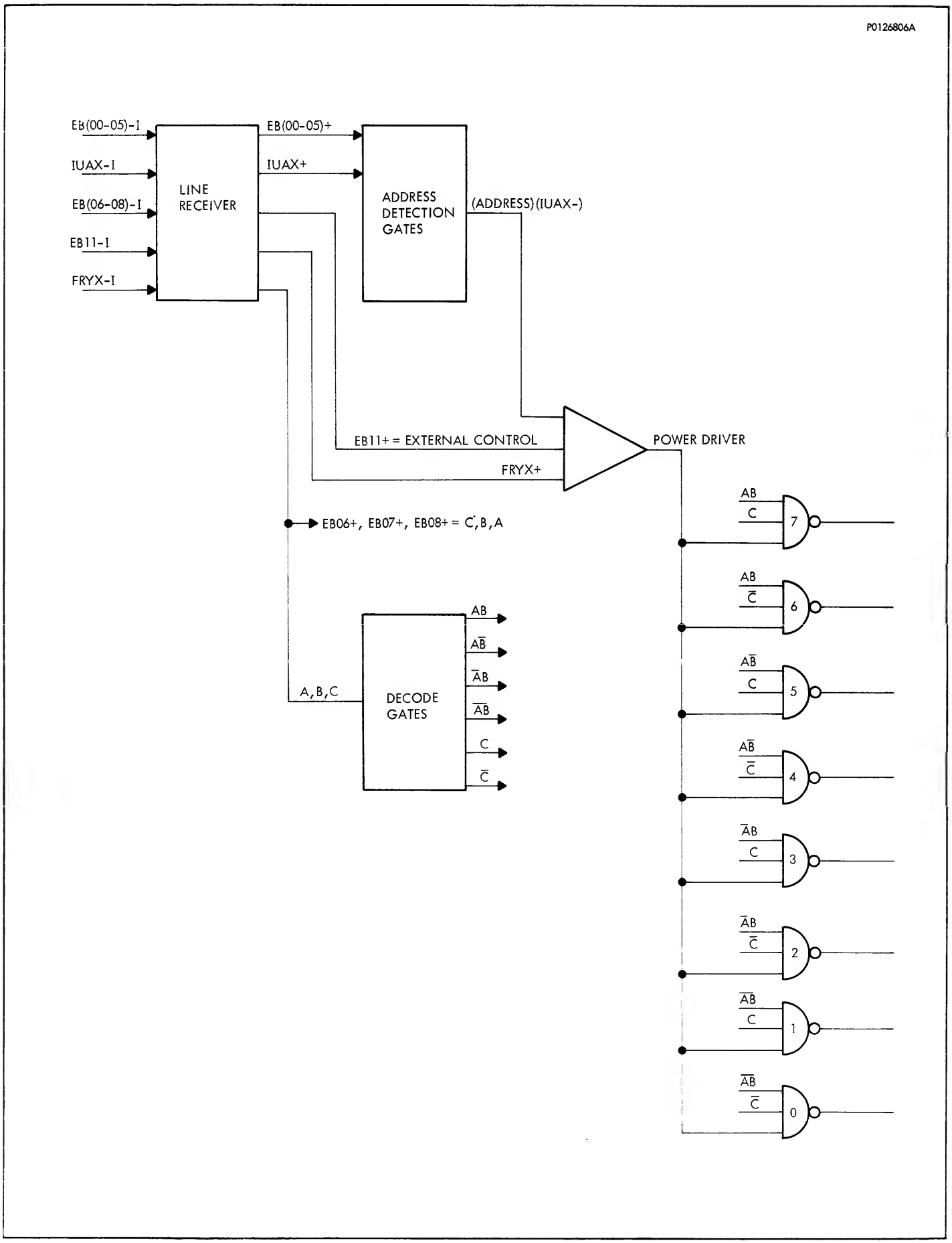
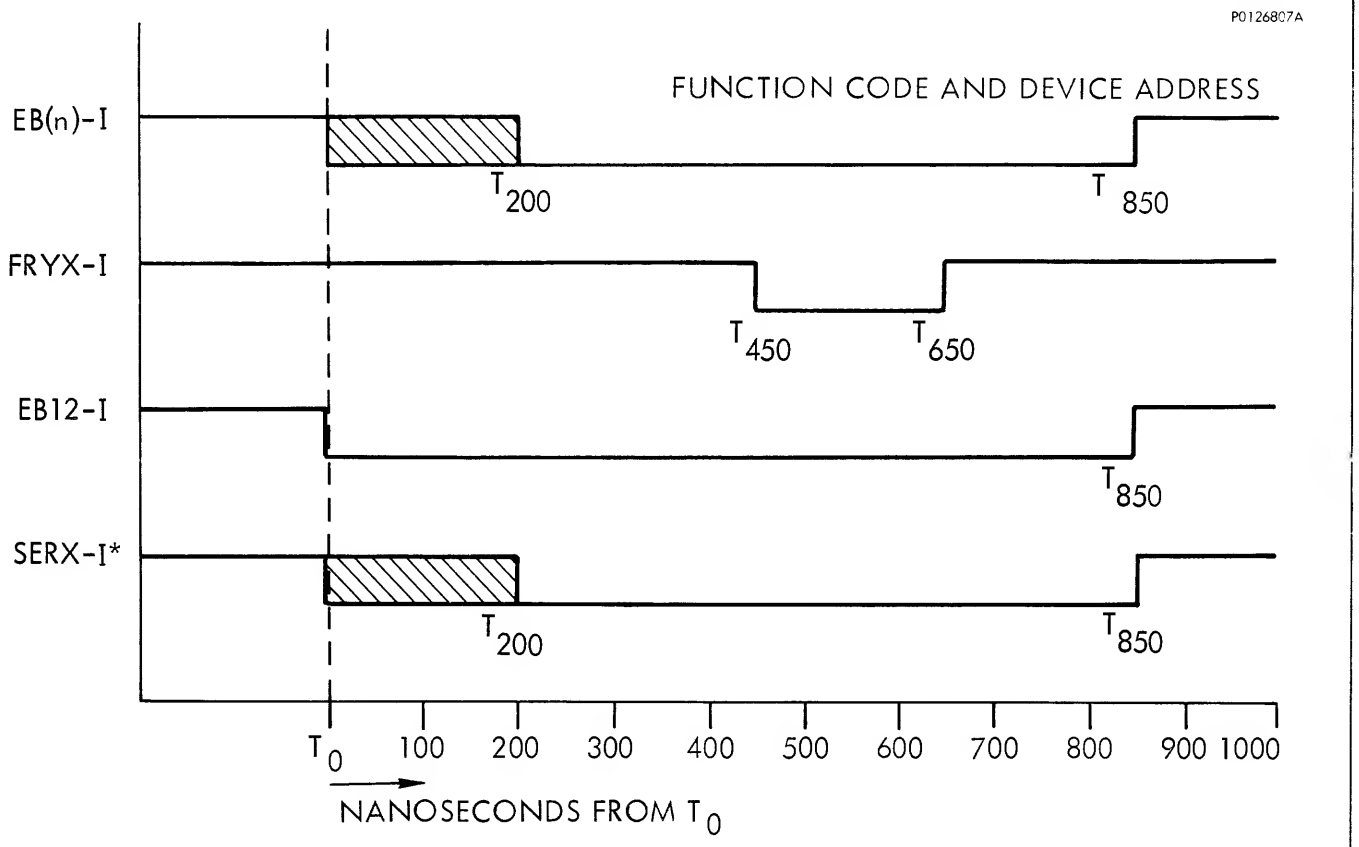
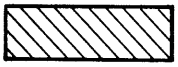


Figure 2-6. Typical Peripheral Controller Logic: EXC Command



T_0 is the start of the execute phase of the sense instruction.

Logic levels: true = 0 vdc,
false = +3 vdc.

 = time when signal is settling.

* SERX-I is normally on at T_{200}; it must be on by T_{650}

Figure 2-7. Sense Response Timing

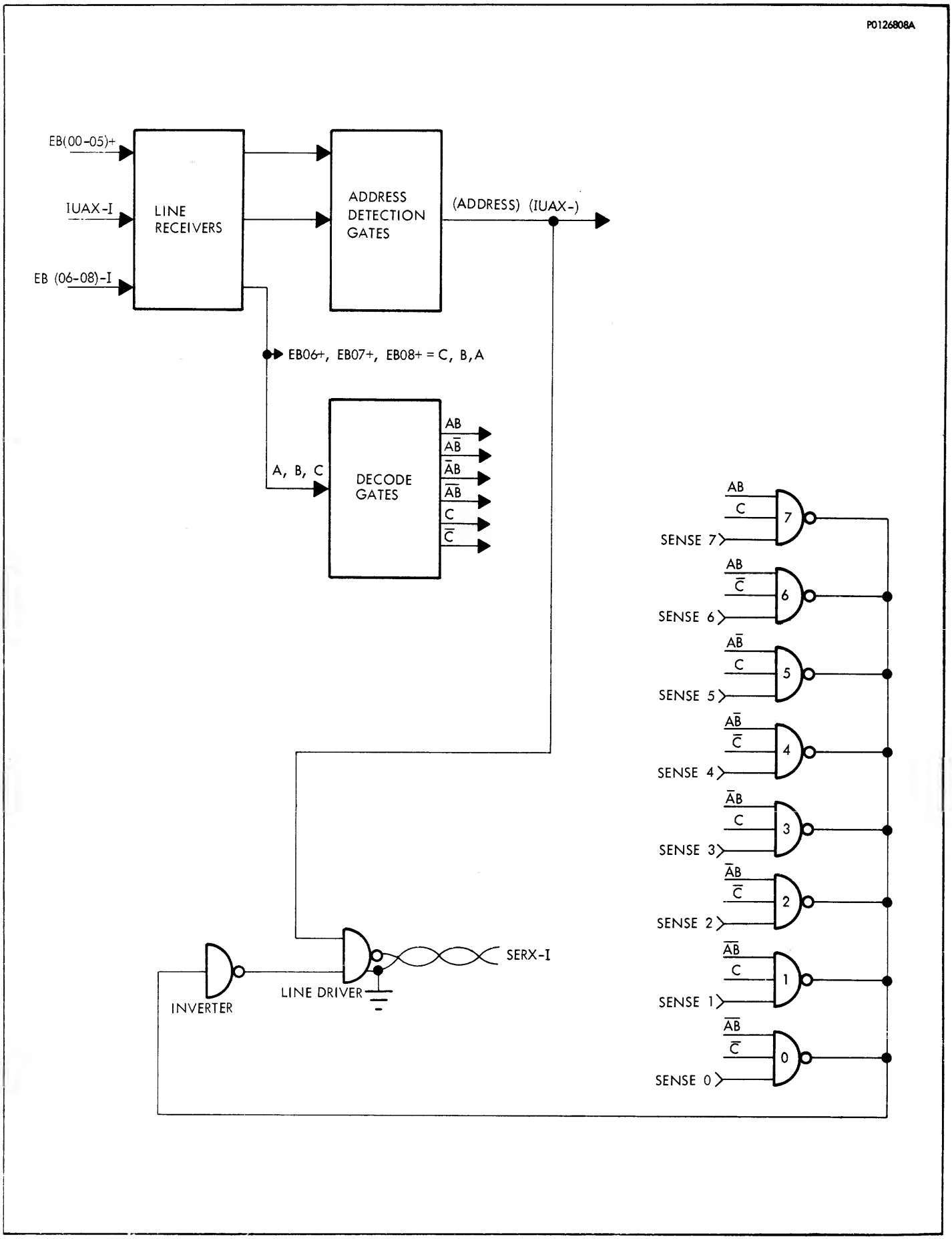
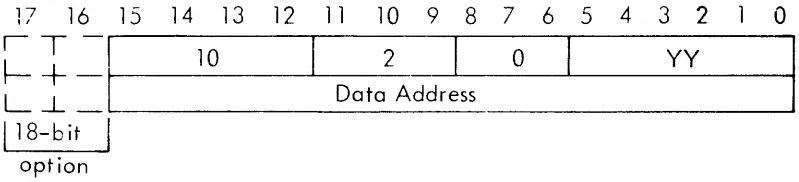


Figure 2-8. Typical Peripheral Controller Logic: SEN Command

The instruction word format for IME includes a second word that specifies the input data word destination.



The execution of any one of the five single-word input transfer commands produces the same sequence of operations on the I/O bus. Consequently, in responding to the bus signals, a peripheral device controller makes no distinction between the input transfer commands.

The execution of an input transfer command is a two-phase operation. The first phase selects the peripheral device controller that will participate in the second-phase data transfer. The transfer timing is shown in figure 2-9. The first phase is initiated by the computer, which places the device address on E-bus lines EB00-I through EB05-I. EB13-I is held true during this phase to indicate that an input transfer command is in progress. Since the E bus is time-shared, a flip-flop in the peripheral controller for the selected device is set to indicate that the controller was selected and that data are to be transferred to the computer. This flip-flop, data transfer in (DTIX+), is set at the true-to-false transition (trailing edge) of FRYX-I (if the controller controls more than one device, an additional flip-flop for each device is required to identify the device selected). As the computer removes the device address and control code information, the selected controller uses DTIX+ to enable the input data onto the E bus. The controller must enable data from the selected device onto the bus no later than 850 nanoseconds after the trailing edge of FRYX-I to strobe the input data. The controller uses the trailing edge of DRYX-I to reset DTIX+, thus removing the input data from the E bus.

When the computer is transferring I/O data under program control, the transfers must be synchronized with the communicating peripheral controller. This synchronization is accomplished by sampling the state of the controller for a ready condition by issuing a sense command prior to the data transfer command.

Figure 2-10 shows typical implementation of the logic required in a peripheral controller to perform an input data transfer.

2.3.4 Single-Word Output Transfer

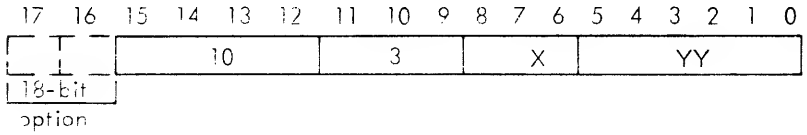
There are three single-word output commands. These are:

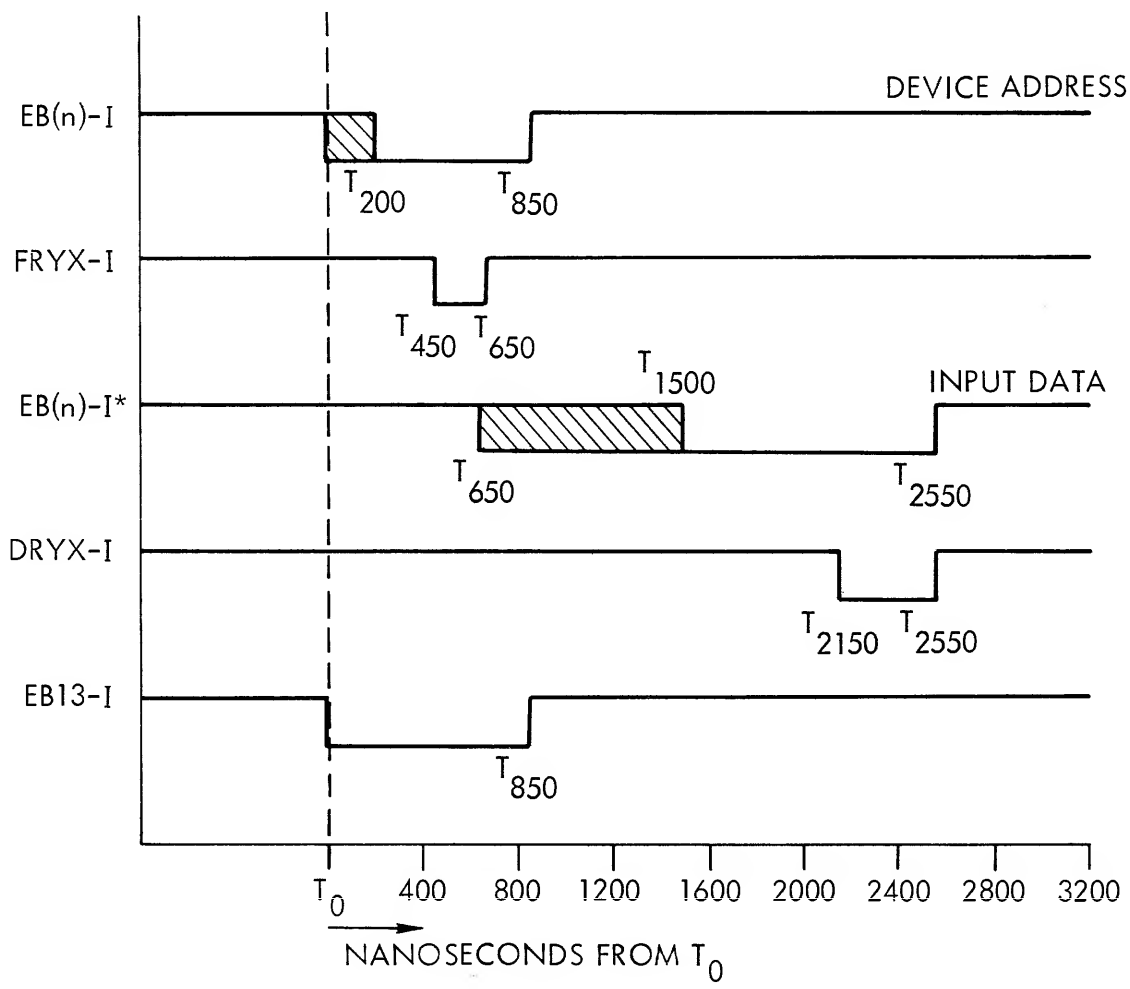
a. Output from A register (OAR).

b. Output from B register (OBR).

c. Output from memory (OME).

The instruction word format for OAR and OBR is shown below, where YY contains the device address and X distinguishes between the two commands.





T_0 is the start of the execute phase of the data transfer in instruction.

Logic levels: true = 0 vdc,
false = +3 vdc.

= time when signal is settling.

* EB(n)-I (input data) must be off by T_{2700}.

Figure 2-9. Data-Transfer-In Timing

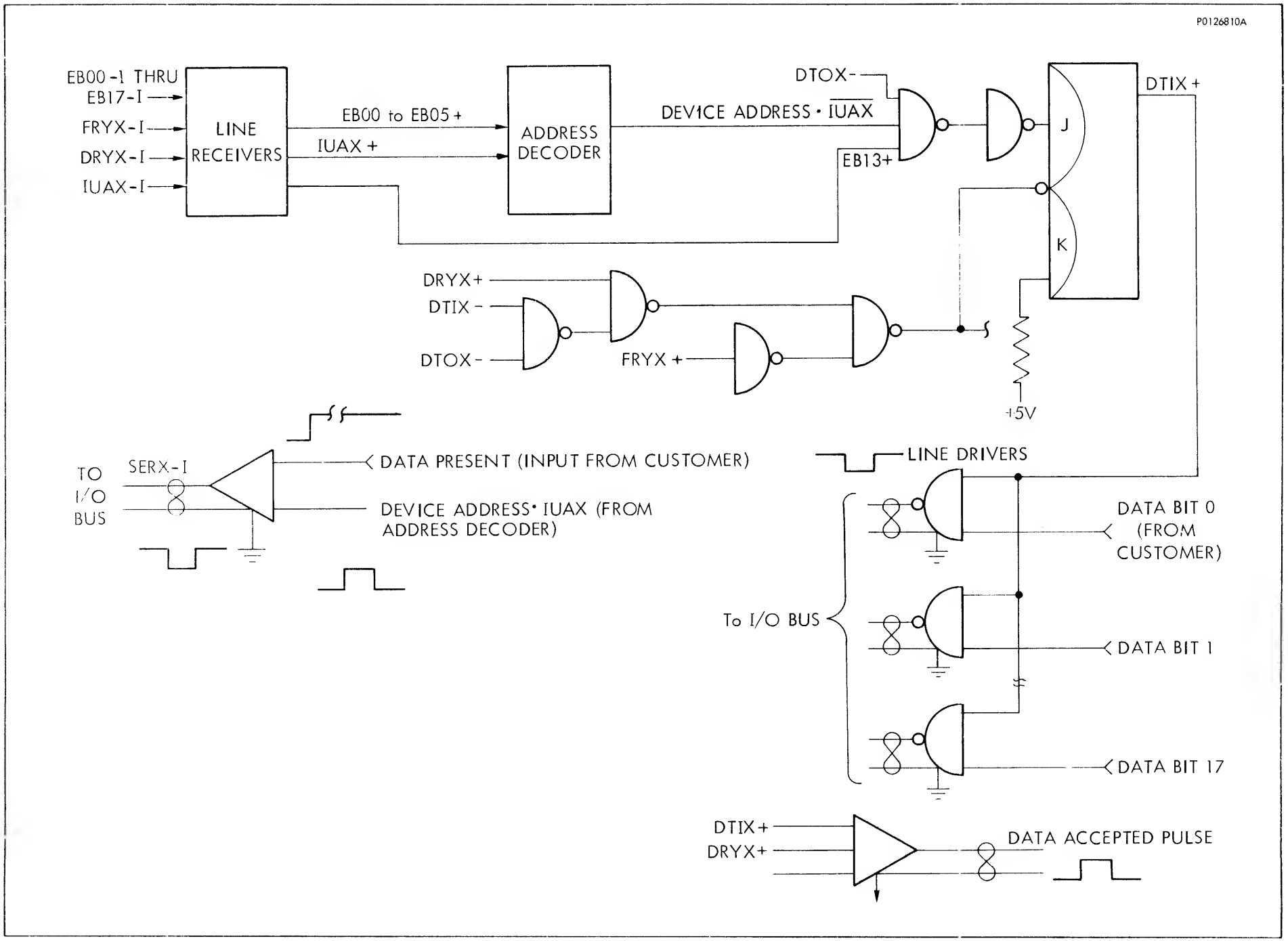
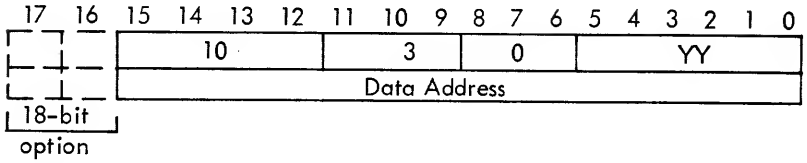


Figure 2-10. Typical Peripheral Controller Logic: Input Data Transfer

This instruction word format for OME (shown below) includes a second word that specifies the output data-word source location.

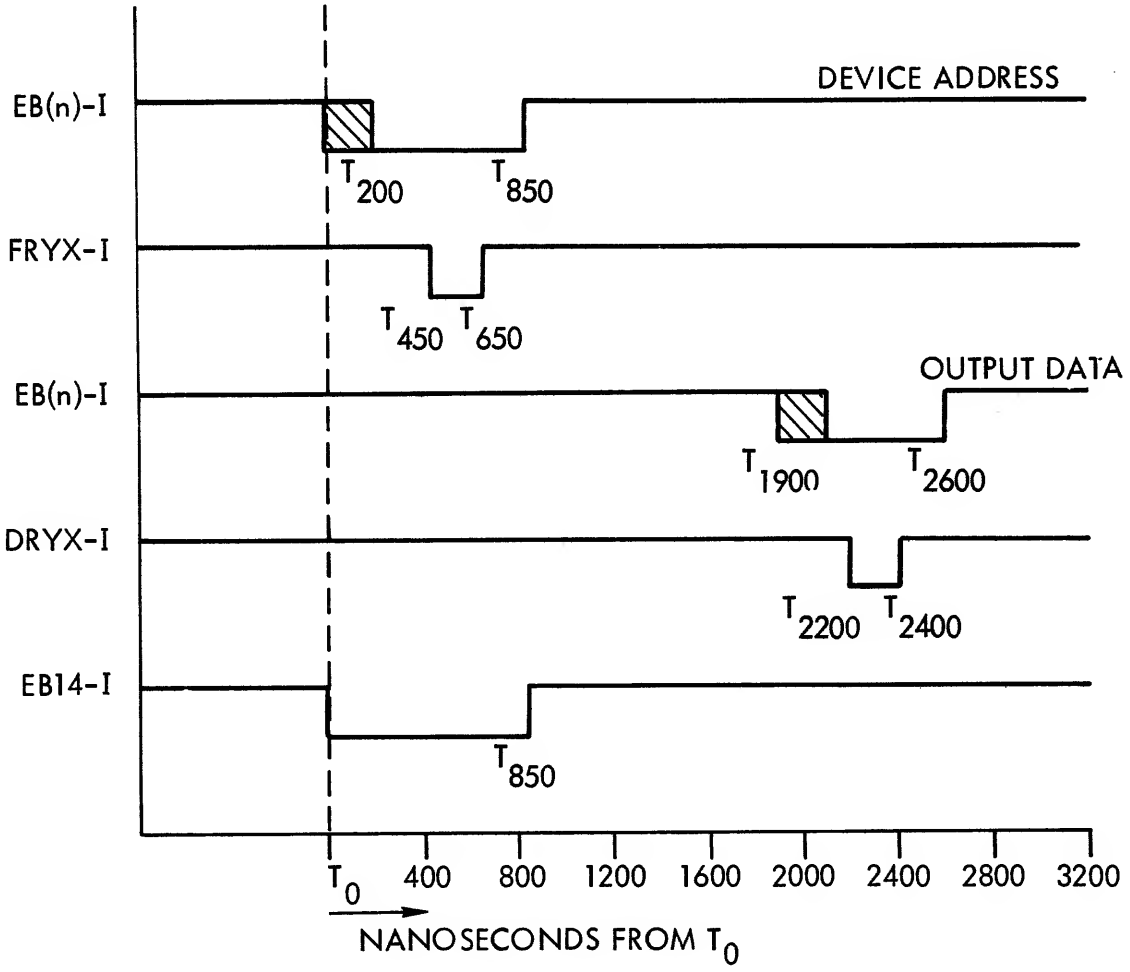


The sequence of signals placed on the I/O bus is the same for each of the three output commands, and the participating peripheral device controller makes no distinction between the output commands.

The execution of an output command is a two-phase operation similar to that described for an input command. The first phase selects the peripheral device which will participate in the second-phase data transfer. The transfer timing is shown in figure 2-11. The first phase is initiated by the computer, which places the device address on E-bus lines EB00-I through EB05-I. EB14-I is held true during this phase to indicate that an output transfer command is being executed. A flip-flop, data transfer out (DTOX+), in the controller for the selected device is set at the trailing edge of FRYX-I (if the controller controls more than one device, an additional flip-flop for each device is required to identify the device selected). Following FRYX-I, the computer removes the device address and control code, and places the output data on the E-bus lines. DTOX+ is used by the controller to gate the contents of the E bus into an input buffer at the trailing edge of DRYX-I. The trailing edge of DRYX-I is also used to reset DTOX+.

When the computer is transferring I/O data under program control, the transfers must be synchronized with the communicating peripheral controller. This synchronization is accomplished by sampling the state of the controller for a ready condition by issuing a sense command prior to the data transfer command.

Figure 2-12 shows typical implementation of the logic required in a peripheral controller to perform an output data transfer.



T_0 is the start of the execute phase of the data transfer out instruction.

Logic levels: true = 0 vdc,
false = +3 vdc.

= time when signal is settling.

Figure 2-11. Data-Transfer-Out Timing

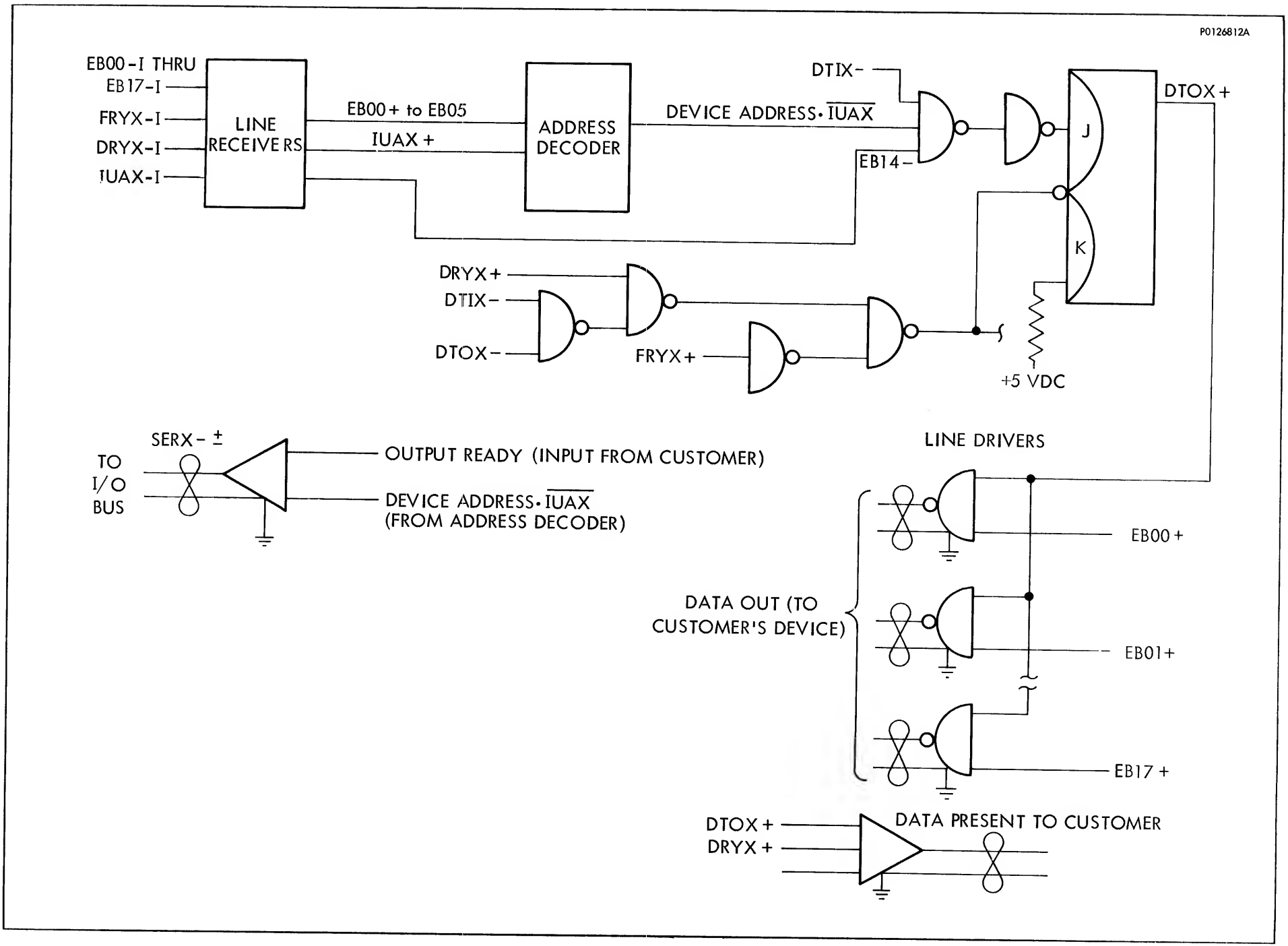


Figure 2-12. Typical Peripheral Controller Logic: Output Data Transfer

SECTION 3
DIRECT-MEMORY-ACCESS-AND-INTERRUPT OPTION

3.1 GENERAL DESCRIPTION

The direct-memory-access-and-interrupt (DMA/I) option of the DATA 620/i provides the control necessary to perform automatic (non-programmed) data transfers to and from memory, and to perform automatic interrupts. This option is required for use with the buffer-interlace-controller option, the power fail/restart option and the priority interrupt option. These options are described in separate reference manuals. The DMA/I may also be used to control automatic data transfers and interrupts when using specially designed peripheral controllers.

3.2 DIRECT-MEMORY-ACCESS FUNCTION

The DMA function provides the capability of data transfers to and from memory without program intervention. It allows a peripheral device option to request the transfer of one word of data while temporarily halting the processing of the stored program. This transfer operation is referred to as trapping. To prevent several peripheral device options from requesting data transfers at the same time, a hard-wired priority scheme determines the order in which the requests may occur. During the interval in which the transfer takes place, the stored program is stopped for 3.15 microseconds (see section 4 for detailed timing of the transfer). This cycle stealing does not disturb the contents of the operational registers (A, B, X, P). In this way, the program may proceed normally at the conclusion of the transfer. With the DMA feature and a controller designed to properly interface with it (such as a buffer interlace controller), the data-transfer rate can be as high as 202,000 words per second, as follows:

DMA trapping sequence:	2.7 microseconds
Synchronization time:	0.45 microseconds
Minimum instruction time:	1.8 microseconds
Period of transfer rate:	4.95 microseconds

Rate = 1/4.95 = 202,000 words per second.

3.3 INTERRUPT FUNCTION

The interrupt function of the DMA/I option provides the capability for peripheral device option to request the computer to execute an instruction independent of the stored computer program. In order to prevent several peripheral device options from requesting interrupts at the same time, a hard-wired priority scheme, which is also used for the DMA function, determines the order in which the requests may occur (such requests will normally originate in priority interrupt modules). During the interrupt, the computer is directed to the memory location specified by the interrupting device and is caused to execute the instruction found at that location. The execution of any instruction other than an I/O command may be requested. Normally, the instruction is a jump-and-mark command and results in the processing of an I/O subroutine. In this event, the interrupt system is automatically inhibited until the inhibit is terminated under program control.

3.4 DIRECT-MEMORY-ACCESS-AND-INTERRUPT CONTROL LINES

The following control lines, along with all of the standard E-bus controls, are used with the DMA/I options: IUAX-I, IURX-I, TPOX-I, TPIX-I, IUCX-I, and IUJX-I. These control lines are logically true at 0 vdc and logically false at +3 vdc. A total of ten receivers or drivers may be connected to each control line.

3.4.1 IUAX-I (Interrupt Acknowledge)

This control signal is set true by the DATA 620/i to acknowledge the receipt of an interrupt, a trap in or a trap out. The interrupting or trapping device controller can communicate an address to the computer and can receive data from or send data to the computer only when this control signal is true. IUAX-I is also used to inhibit device address decoding in every device controller during the address phase of an interrupt or DMA operation. This prevents the controllers from interpreting the lower-order bits of a memory address as a device address.

3.4.2 IURX-I (Interrupt Request)

By setting IURX-I true, the interrupting device controller requests the DATA 620/i to execute an instruction. The address of this instruction is placed on the E bus by the interrupting device upon receipt of IUAX-I from the computer.

3.4.3 TPOX-I (Trap-Out Request)

By setting TPOX-I true, the trapping device controller requests the DATA 620/i to output one word of data from memory. The address of this word is placed on the E bus by the controller upon receipt of IUAX-I from the computer.

3.4.4 TPIX-I (Trap-In Request)

By setting TPIX-I true, the trapping device controller requests the DATA 620/i to input one word of data to memory. The address of this word is placed on the E bus by the controller upon receipt of IUAX-I from the computer.

3.4.5 IUCX-I (Interrupt Clock)

This control signal is a 1.1-MHz clock from the DATA 620/i that is disabled by IUAX-I. When IUAX-I is false, the clock is present on the IUCX-I line. When IUAX-I is true, IUCX-I is held true. The true-to-false transition of IUCX-I is used to set the request flip-flops (IURX-I, TPOX-I, TPIX-I) in respective controllers.

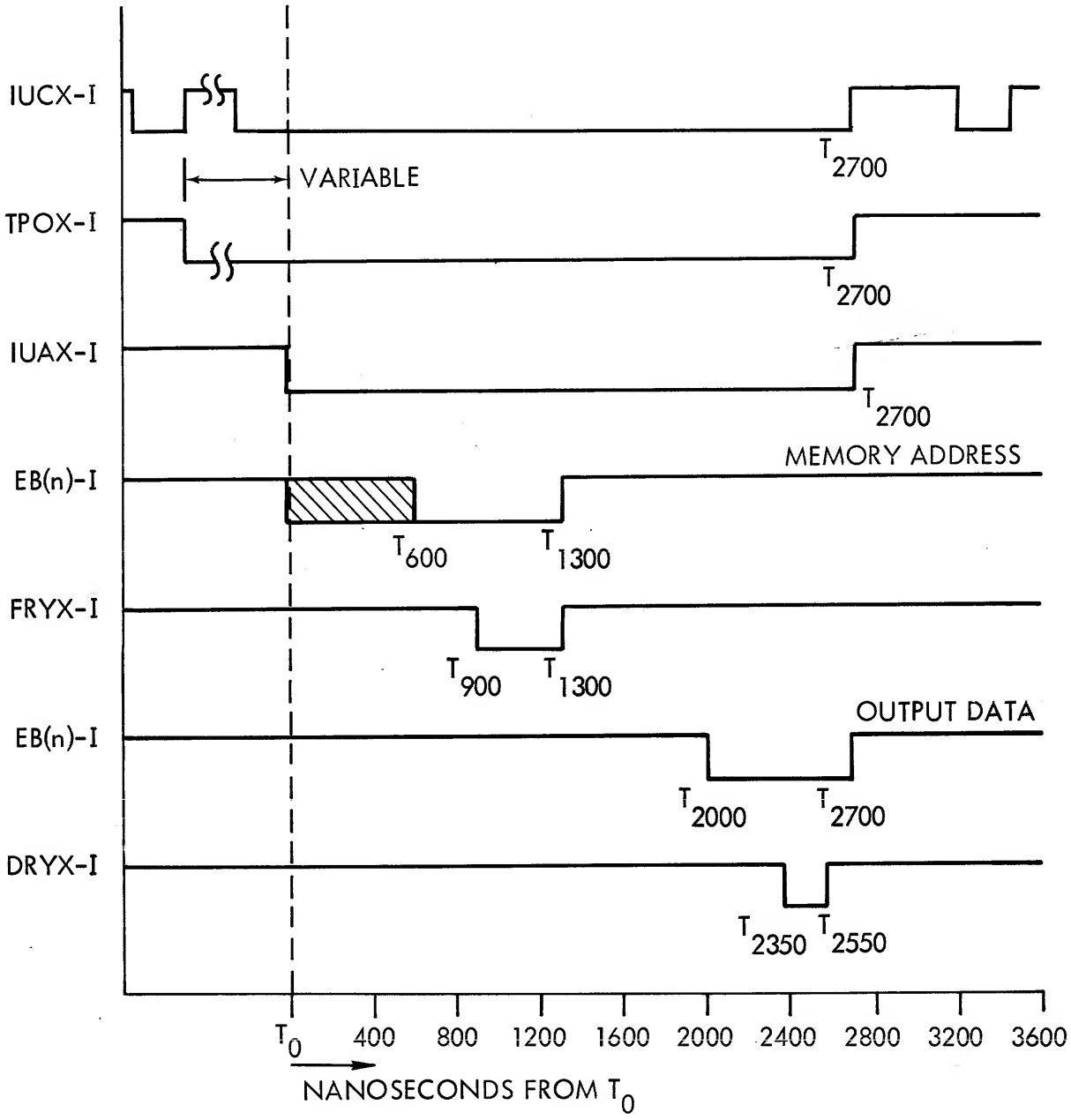
3.4.6 IUJX-I (Interrupt Jump)

This control signal from the computer is used to inhibit all interrupts that occur after a jump-and-mark instruction when that instruction is the result of an interrupt request. This signal becomes true 2.7 microseconds from the false-to-true transition of IUAX-I and remains true for 450 nanoseconds.

3.5 DETAILS OF DIRECT-MEMORY-ACCESS SEQUENCE

A controller using the DMA option must be able to generate or respond to the following signals: IUAX-I, TPOX-I, TPIX-I, IUCX-I, FRYX-I, DRYX-I, and EBNX-I, plus priority signals on the PRMX-I input and on the PRNX-I output.

The DMA (trap in/out) sequence is comprised of three phases: request, address and data, as shown in the DMA timing sequence, figures 3-1 and 3-2.



T_0 is the start of the output sequence.

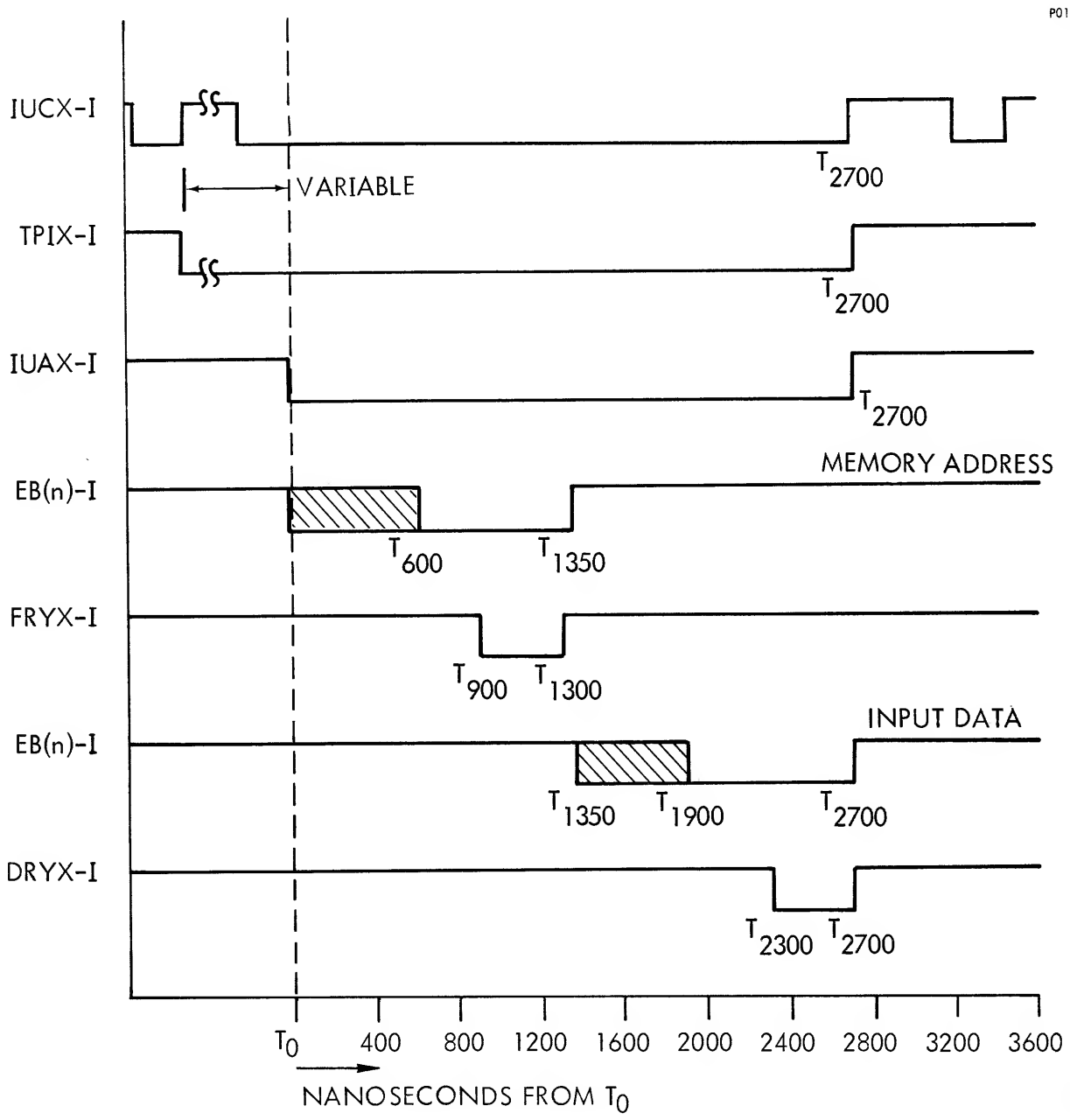
Logic levels: true = 0 vdc,
false = +3 vdc.

= time when signal is settling.

TPOX-I is normally off at T_{2700}; it must be off by T_{2900}.

EB(n)-I (memory address) is normally on at T_0; it must be on by T_{600}. It is normally off at T_{1300}; it must be off by T_{1500}.

Figure 3-1. Trap-Out Timing



T_0 is the start of the input sequence.

Logic levels: true = 0 vdc,
false = +3 vdc.

TPIX-I is normally off at T_{2700}; it must be off by T_{2900}.

EB(n)-I (memory address) is normally on at T_0; it must be on by T_{600}. It is normally off at T_{1300}; it must be off by T_{1600}.

EB(n)-I (input data) is normally on at T_{1350}; it must be on by T_{1900}. It is normally off at T_{2700}; it must be off by T_{2900}.

Figure 3-2. Trap-In Timing

3.5.1 Request Phase

The peripheral controller of the requesting device option having the highest priority generates a trap request (sets TPOX-I or TPIX-I true) on the trailing edge of the interrupt clock (IUCX-I) and waits for an interrupt acknowledge (IUAX-I) from the computer.

3.5.2 Address Phase

Interrupt acknowledge (IUAX-I) is received by the peripheral controller of the requesting device, thereby holding IUCX-I true. The controller then places the memory address, into which or from which data are to be read, onto the E bus and waits for the trailing edge of FRYX-I.

3.5.3 Data Phase

At the trailing edge of FRYX-I, the controller removes the address from the E bus. Data are then placed on the E bus by either the computer or the requesting device controller. These data are gated into the device or into the computer at the trailing edge of DRYX-I. All signals are removed from the bus when IUAX-I becomes false.

3.6 DETAILS OF INTERRUPT

A controller communicating with the interrupt portion of the DMA/I option must be able to generate or respond to the following signals: IUAX-I, IUCX-I, IURX-I and EBNX-I, plus priority signals on the PRMX-I input and on the PRNX-I output. The interrupt sequence is comprised of two phases: request and address, as shown in the interrupt timing sequence, figure 3-3.

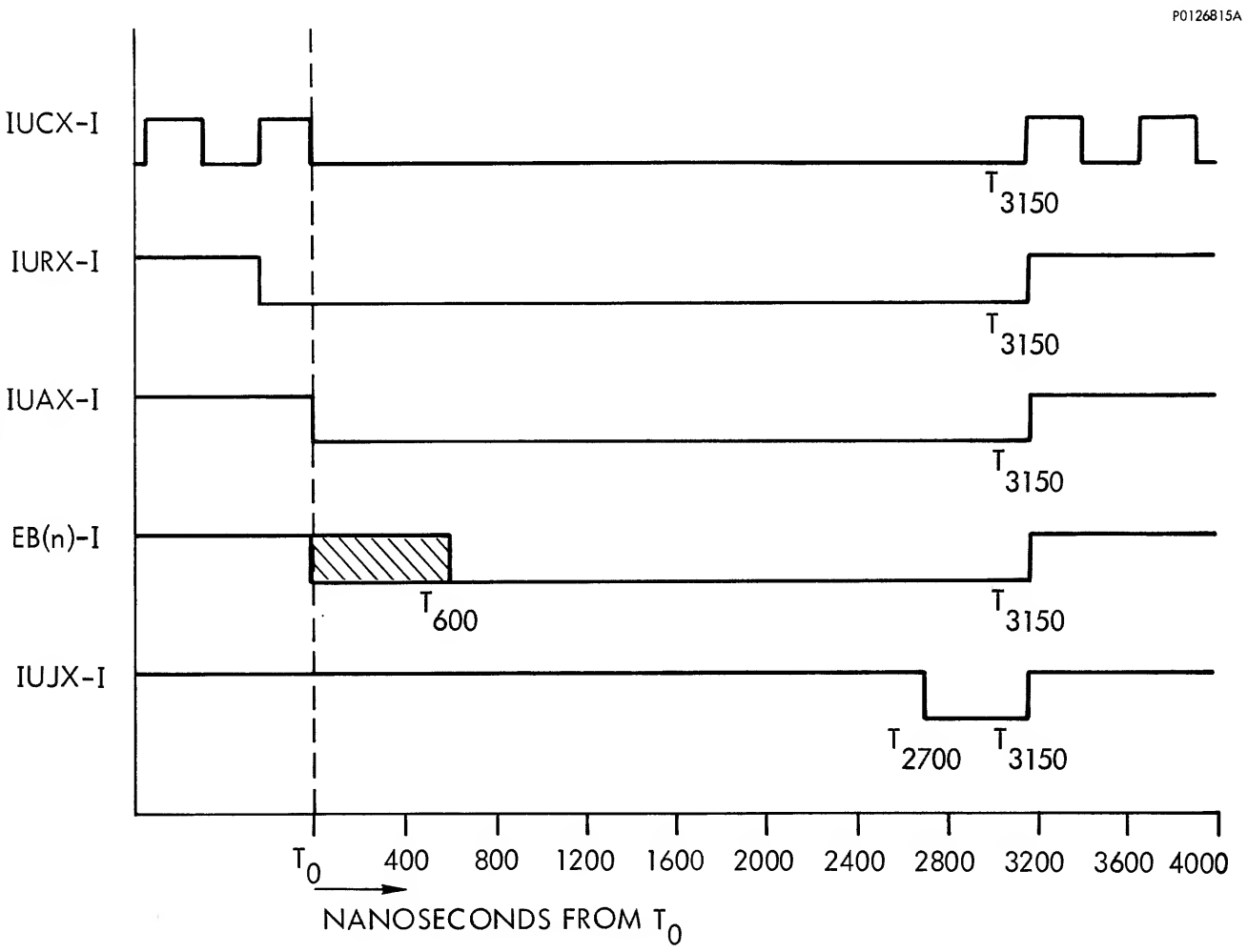
3.6.1 Request Phase

The peripheral controller of the requesting device option having the highest priority generates an interrupt request (sets IURX-I true) on the trailing edge of the interrupt clock (IUCX-I). The controller then waits for an interrupt acknowledge (IUAX-I) from the computer.

3.6.2 Address Phase

When the DATA 620/i has recognized the request, it responds by setting IUAX-I true, thus holding IUCX-I true. Upon receipt of IUAX-I from the computer, the requesting device controller places on the E bus the address containing the instruction to be executed. The instruction at this address may be any instruction except an I/O instruction. In the case of a two-word instruction, the DATA 620/i ORs a binary ONE into bit position zero to form the address of the second word, making this address an odd number.

Because of this, the address placed on the E bus by the requesting device controller must be an even number. If the instruction is a jump-and-mark instruction, the controller receives an interrupt-jump signal (IUJX-I) from the computer. IUJX-I resets the master enable in the controller of every device capable of making a request, thereby inhibiting further interrupts from these controllers during the subsequent subroutine. Each controller must be re-enabled by the computer program at the end of the subroutine. The address phase ends when IUAX-I becomes false. All signals must be removed from the bus at this time.



T_0 is the start of the timing and interrupt sequence.

Logic levels: true = 0 vdc,
false = +3 vdc.

▨ = time when signal is settling.

IURX-I is normally off at T_{3150}; it must be off by T_{3300}.

EB(n)-I is normally on at T_0; it must be on by T_{600}. It is normally off at T_{3150}; it must be off by T_{3300}.

IUJX-I is present for a jump and mark instruction.

Figure 3-3. Interrupt Timing

3.7 EXTERNAL DEVICE PRIORITY

Every peripheral device option on the I/O bus which is capable of automatic requests (interrupt, trap in and trap out) must be assigned a level of priority. This is to assure that only one such option at a time will be initiating an automatic request. Only options capable of initiating automatic requests require a priority assignement.

The priorities assigned determine the electrical location of each peripheral device option on the I/O bus. Priority lines PR1X-I, PR2X-I, PR3X-I and PR4X-I in the various I/O cables are used to place the options in the assigned order on the I/O bus. This graphically shown in figures 3-4 and 3-5. The simplest interconnection results when the physical position corresponds directly to the priority assignment as in figure 3-4. In this case, the PR2X-I and PR3X-I lines are not used. When the physical location on the cable does not correspond directly to the priority assignment, a more complex interconnection results, such as in figure 3-5. The controller of each option contains priority logic to determine when that option has priority. Typical priority logic is shown in figure 3-6.

In figure 3-6, IURX-I, TPIX-I and TPOX-I are the interrupt, trap-in and trap-out request signals, respectively. These signals have previously been described.

3.7.1 PRMX-I

PRMX-I is the priority-in input to the controller of each peripheral device option. When this signal is true (0 vdc) for a given option, no option having a higher priority has requested an interrupt or trap. The controller of the given option can then initiate a request if its request flip-flop is set. When PRMX-I is false (+3 vdc) for a given option, an option having a higher priority is performing a request. The controller of the given option is therefore inhibited from initiating a request, regardless of the state of its request flip-flop. If an option having a higher priority initiates a request before a request from a lower-priority option is acknowledged (i.e., before IUAX-I becomes true), the higher-priority option assumes control.

3.7.2 PRNX-I

PRNX-I is the priority-out output from the controller of each peripheral device option. This signal is false (+3 vdc) from the controller of a given option if either the given option or one with higher priority is initiating a request. When false, this signal prevents all lower-priority options from initiating requests.

Figure 3-4 shows typical priority line connection for controllers in the main frame or in the same expansion chassis.

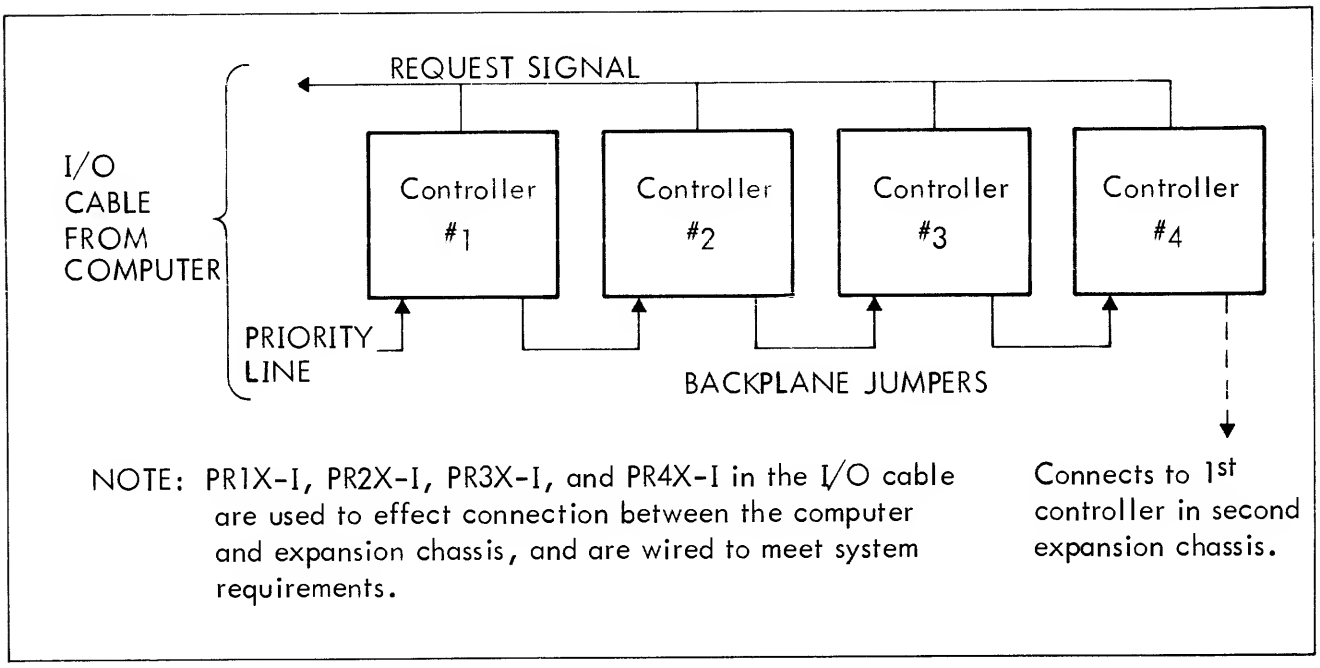


Figure 3-4. Priority scheme for controllers installed in the same expansion chassis.

Controllers which are installed in different expansion chassis, connected through the I/O cable, require use of additional lines (PR2X-I, PR3X-I, and/or PR4X-I). See figure 3-5.

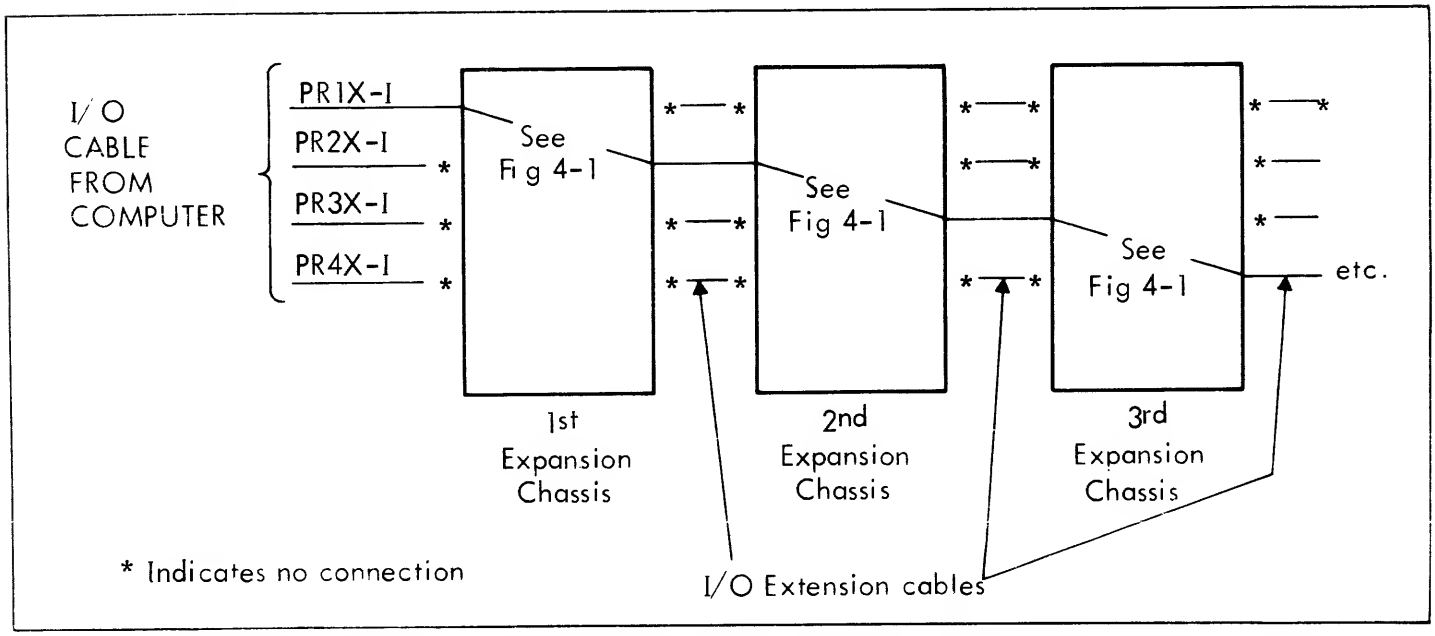


Figure 3-5. Priority Line Connection when more than one expansion chassis is installed.

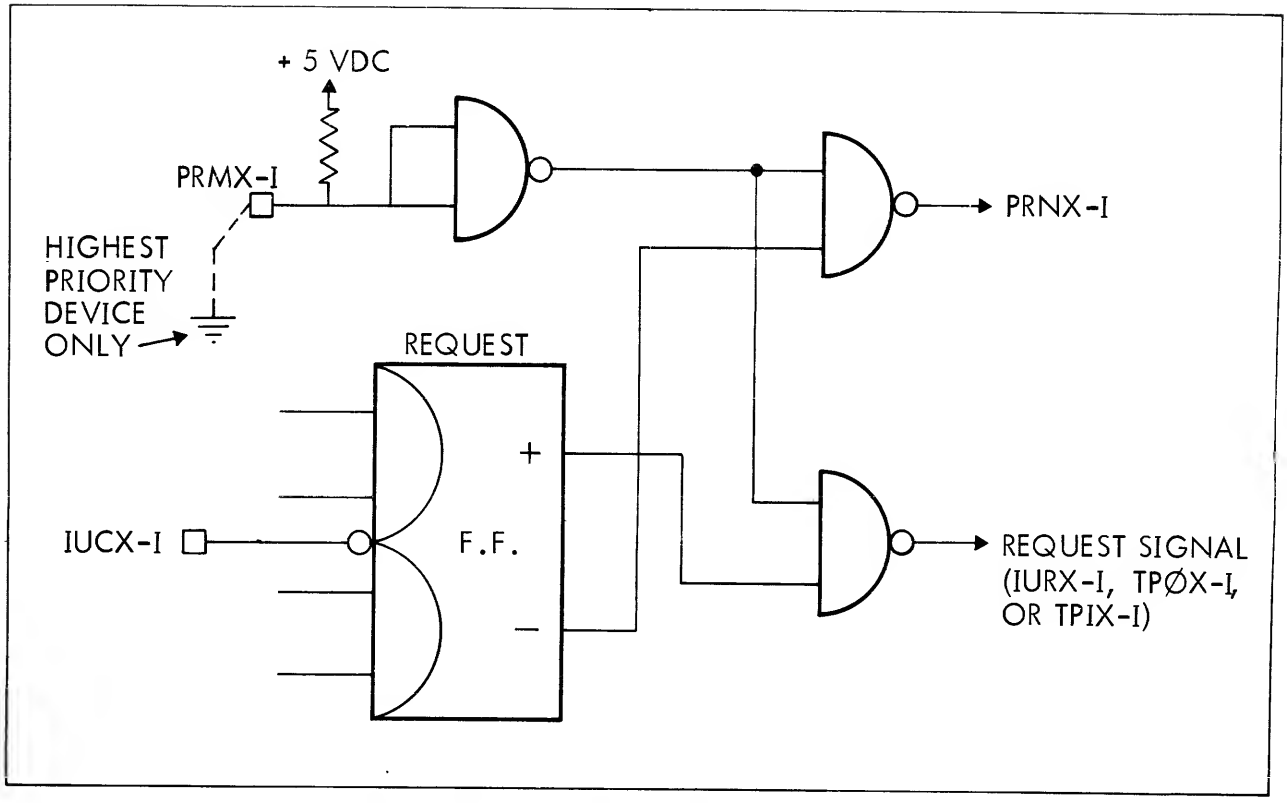


Figure 3-6. Peripheral Device Priority Logic

SECTION 4
I/O CABLE INTERFACE CHARACTERISTICS

4.1 I/O BUS LINE

The I/O-bus interface generally involves three types of driver/receiver configurations. The first configuration is represented by the bidirectional E bus. The time-shared nature of the E bus requires that the computer and most peripheral controllers employ one cable driver and one cable receiver for each line of the bus. A typical E-bus configuration is shown in figure 2-2. E-bus communication is such that information is transferred from the computer cable driver to a peripheral controller cable receiver or from a controller driver to the computer receiver. Communication between controllers does not occur on the E bus. Each bus line is terminated by 150 ohms to +3 vdc at the computer, and in a termination shoe at the opposite end of the I/O cable. Ferromagnetic beads (VDM P/N 15A0005) are used to control the rise and fall times of the E-bus signals. It is necessary to place three beads in the vicinity of the signal driver to insure satisfactory signal control; consequently, beads are placed on the driver outputs as shown in figures 2-2, 2-3 and 2-4.

4.2 I/O CABLE DRIVER AND RECEIVER

The I/O cable driver is a four-input NAND gate of the Texas Instruments SN15-844N series (or equivalent) which has been selected so that the saturated common-emitter output stage is capable of conducting up to 70 milliamperes without exceeding a saturation voltage of 0.5 vdc. This driver can be collector-tied with another driver to produce a negative-logic OR function. The driver circuit schematic is shown in figure 4-1.

The I/O cable receiver is a Texas Instruments series SN15-846N two-input NAND gate (or equivalent). The receiver circuit schematic is shown in figure 4-2. The receiver input represents a maximum load (current source) of 1.4 milliamperes at 0 vdc. The receiver input switching threshold is 1.5 $\pm$ 0.5 vdc.

4.3 I/O CABLE AND CONNECTOR

The I/O cable contains 37 twisted-pair, 24-gauge stranded wires. The I/O cable is composed of several cable segments. The first segment extends from the plug, P32, on the computer to receptacle J32 on the first expansion frame (or the customer chassis). This is shown in figure 2-2, 2-3 or 2-4. The second segment extends from the cable plug, P32, on the first expansion frame to the receptacle, J32, on the second expansion frame. The termination shoe is shown in figure 2-2, 2-3 or 2-4 as being mounted on receptacle J32 on the second expansion frame. When a third expansion frame is added, the termination shoe must be removed, the interconnecting cable attached to J32 and the termination shoe mounted on receptacle J32 of the third expansion frame. Similarly, when special external equipment is connected to the I/O cable, the termination shoe is mounted on the special equipment. The I/O cable connector pin assignments are listed in table 4-1. Total cable length is limited to 20 feet.

P0126819A

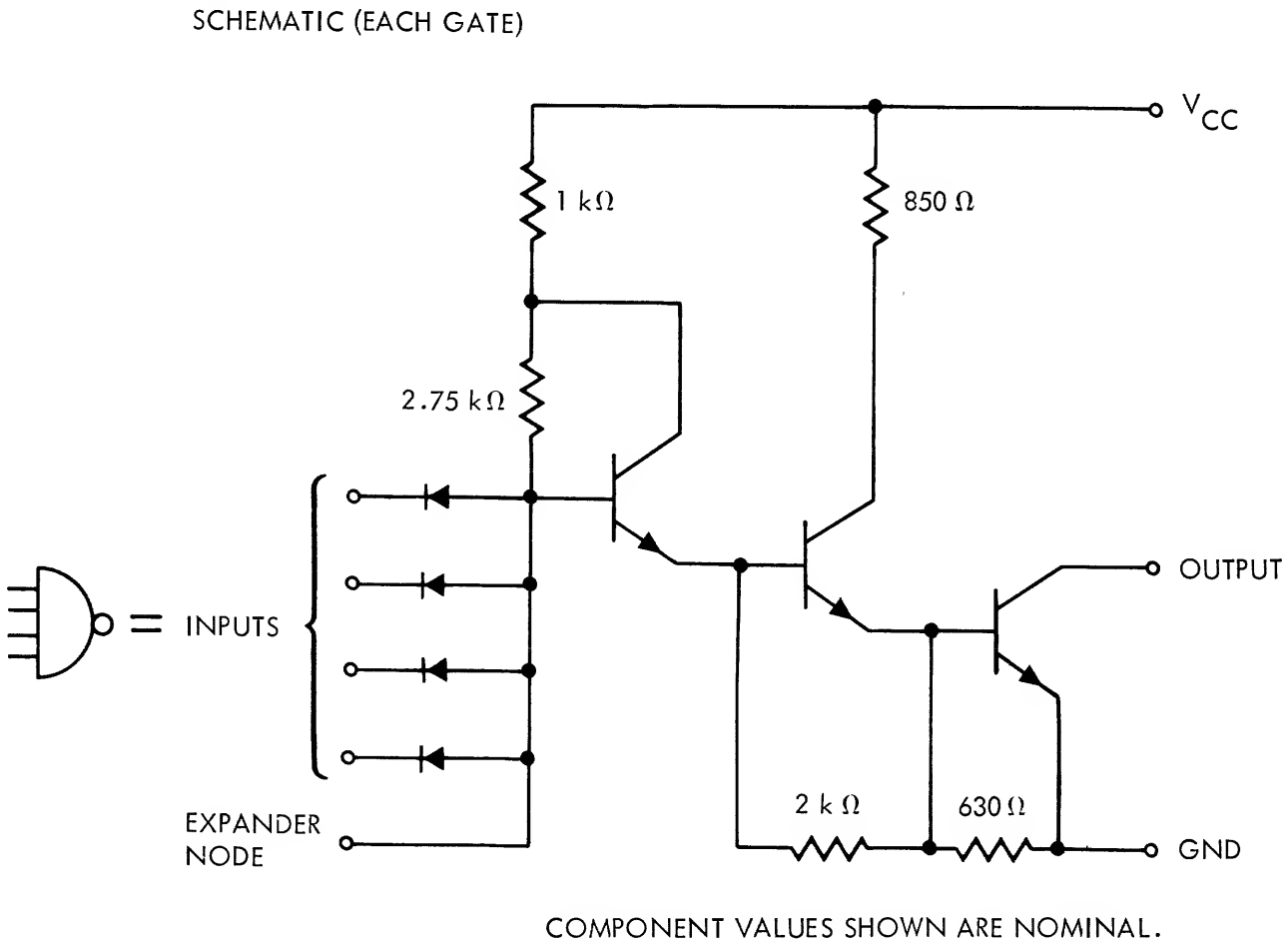
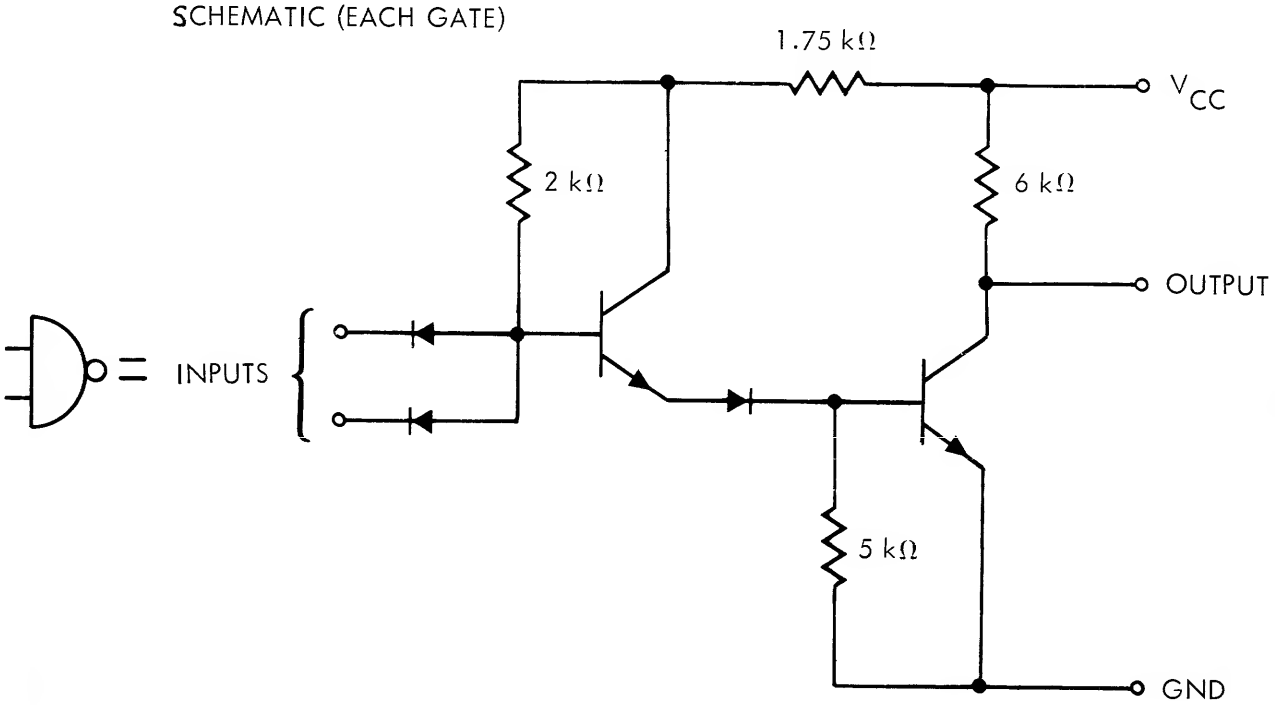


Figure 4-1. I/O Cable Driver

P0126820A



COMPONENT VALUES SHOWN ARE NOMINAL.

Figure 4-2. I/O Cable Receiver

TABLE 4-1
I/O Cable Connector Pin Assignments

Line	Pin	Function
1	1	EB00-I
2	2	R
3	3	EB01-I
4	4	R
5	5	EB02-I
6	7	R
7	8	EB03-I
8	10	R
9	11	EB04-I
10	12	R
11	13	EB05-I
12	14	R
13	15	EB06-I
14	16	R
15	17	EB07-I
16	18	R
17	20	EB08-I
18	21	R
19	22	EB09-I
20	23	R
21	24	EB10-I
22	25	R
23	26	EB11-I
24	27	R
25	28	EB12-I
26	29	R

Line	Pin	Function
27	30	EB13-I
28	31	R
29	32	EB14-I
30	33	R
31	34	EB15-I
32	35	R
33	36	EB16-I
34	37	R
35	38	EB17-I
36	39	R
37	40	FRYX-I
38	41	R
39	42	DRYX-I
40	43	R
41	44	SERX-I
42	45	R
43	46	TPIX-I
44	47	R
45	48	TPOX-I
46	49	R
47	50	PR1X-I
48	51	R
49	52	PR2X-I
50	53	R
51	54	PR3X-I
52	55	R

TABLE 4-1 (continued)
I/O Cable Connector Pin Assignments

Line	Pin	Function
53	56	PR4X-I
54	57	R
55	58	SYRT-I
56	59	R
57	60	IUAX-I
58	62	R
59	63	IUCX-I
60	64	R
61	65	IURX-I
62	66	R
63	67	IUJX-I

Line	Pin	Function
64	70	R
65	71	
66	72	
67	73	
68	74	
69	75	
70	76	
71	77	
72	78	
73	79	
74	80	+3 vdc
85	82	GND